

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**Implantación Y Gestión De Espacios Colaborativos Bajo
Entorno Android, Sistema de Almacenamiento De
Contenido Alfresco E Integridad De La Información
Mediante Firma Digital**



AUTOR: Javier Alpañez Soler
DIRECTOR: Juan Carlos Sánchez Aarnoutse
Septiembre 2012



Autor	Javier Alpañez Soler
E-mail del Autor	javier_alpenez@hotmail.com
Director	Juan Carlos Sánchez Aarnoutse
E-mail del Director	juanc.sanchez@upct.es
Codirector(es)	
Título del PFC	Implantación Y Gestión De Espacios Colaborativos Bajo Entorno Android, Sistema de Almacenamiento De Contenido Alfresco E Integridad De La Información Mediante Firma Digital
Descriptores	
<p>Resumen</p> <p>Desarrollo de una aplicación para dispositivos Android que sea capaz de manejar Espacios de Colaboración entre varios usuarios para el acceso a documentación o ficheros comunes. Además debe ser capaz de soportar diferentes formas de acceso a la aplicación mediante permisos así como múltiples Espacios para cada usuario.</p> <p>Se reforzarán los documentos compartidos usando para su almacenamiento un gestor documental que sea capaz de gestionar y soportar las acciones que por parte del cliente se realicen.</p> <p>Por último, se integrará el concepto de firma digital para asegurar la integridad de los documentos que se suben por un usuario. Se utilizará para ello una estructura de firma electrónica que sea reconocida por el marco legal vigente.</p>	
Titulación	Ingeniero en Telecomunicación
Intensificación	
Departamento	Tecnologías de la Información y Comunicaciones
Fecha de Presentación	septiembre - 2012

Agradecimientos

En memoria de mis abuelos, los cuales estarían muy orgullosos de este día.

Índice General

Introducción.....	15
1.1 Motivación.....	15
1.2 Objetivos del Proyecto	16
1.3 Estructura de la memoria.....	16
 Contexto / Tecnologías	17
2.1 S.O. Android.....	17
2.1.1 ¿Qué es Android?	17
2.1.2 Libertad.....	18
2.1.3 Un poco de historia.....	18
2.1.4 Desarrollo y Estructura de las Aplicaciones.....	19
2.1.5 Componentes de una aplicación Android.....	20
2.2 Gestor Documental Alfresco	21
2.2.1 Arquitectura.....	21
2.3 Firma Digital	22
2.3.1 XadES.....	22
2.3.2 Estructura XadES Detached	22
2.4 Base de Datos MySql	24
2.5 Servicios Web.....	24
2.5.1 Protocolo SOAP	25
2.6 Hibernate	25
2.7 Herramientas Utilizadas Desarrollo.....	26
2.7.1 Eclipse + Android SDK.....	26
2.7.2 Toad for MySql	27

Implementación de la Aplicación	28
3.1 Objetivos de la implementación	28
3.2 Arquitectura del Proyecto	29
3.3 Aplicación Cliente	30
3.3.1 Estructura de la Aplicación.....	30
3.3.2 Implementación Clases - Directorio src	30
3.3.2.1 es.upct.pfc.adapters	31
3.3.2.2 es.upct.pfc.eupct	31
3.3.2.3 es.upct.pfc.listeners	32
3.3.2.4 es.upct.pfc.notifications	33
3.3.2.5es.upct.pfc.task	33
3.3.2.6 es.upct.pfc.utils	34
3.3.3 Implementación de Interfaz - Directorio res	35
3.3.3.1 Drawable.....	35
3.3.3.2 Layout.....	36
3.3.3.3 Menu	37
3.3.3.4 Values	38
3.3.4 Manifiesto de la aplicación.....	38
3.4 Servidor	39
3.4.1 Servicios Web.....	39
3.4.1.1 Estructura del Servidor de Servicios Web	40
3.4.1.2 es.upct.pfc.ws	41
3.4.1.3 es.upct.pfc.alfresco	41
3.4.1.4 es.upct.pfc.hibernate	41
3.4.1.5 es.upct.pfc.beans.....	42
3.4.1.6 es.upct.pfc.utils	42
3.4.1.7 alfresco	43
3.4.1.8 Hibernate.cfg.xml	43
3.4.2 Alfresco	44

3.4.2.1 Estructura del Gestor Documental.....	44
3.4.2.2 Puesta en marcha	45
3.4.3 MySql	45
3.4.3.1 Estructura de la Base de Datos	46
3.4.3.2 Puesta en marcha	47
Manual de Usuario	49
4.1 Aplicación Cliente	49
4.1.1 Instalación / Desinstalación de la Aplicación.....	49
4.1.2 Ejecución de la Aplicación. Casos de Uso.	51
4.1.3 Funcionalidades Generales	51
4.1.3.1 Presentación.....	52
4.1.3.2 Acceso a la Aplicación por Usuario y Contraseña	52
4.1.3.3 Acceso con Certificado Digital	53
4.1.3.4 Navegación por los Espacios Colaborativos	54
4.1.3.5 Visualización de Ficheros.....	55
4.1.3.6 Visualización del Fichero de Firma Digital.....	56
4.1.3.7 Descarga de Ficheros a Tarjeta SD.....	58
4.1.3.8 Notificaciones.....	59
4.1.3.9 Menú Información	61
4.1.3.10 Salir de la Aplicación	61
4.1.4 Funcionalidades sólo para Administradores.....	62
4.1.4.1 Creación Carpetas.....	62
4.1.4.2 Subida Ficheros	64
4.1.4.3 Subida Y Firma de Ficheros	65
4.1.4.4 Borrar Ficheros / Carpetas	67
4.1.4.5 Renombrado de Ficheros / Carpetas.....	68
4.2 Administración Servidor.....	70
4.2.1 Ejecución de los Servicios. Casos de Uso.	70

4.2.2 Alta de un Usuario.....	70
4.2.3 Creación de un nuevo Espacio de Colaboración	72
4.2.4 Asignación de Permisos a un Usuario en un Espacio.....	76
Conclusión y Líneas Futuras	80
Bibliografía.....	82

Índice de Figuras

Figura 2.1 Estructura Proyecto Android.....	19
Figura 2.2 Eclipse + Android SDK	27
Figura 2.3 Toad For MySql	27
Figura 3.1 Arquitectura Proyecto	29
Figura 3.2 Estructura Aplicación Cliente	30
Figura 3.3 Estructura Carpeta Src.....	31
Figura 3.4 Estructura Carpeta Adapters	31
Figura 3.5 Estructura Carpeta Principal	32
Figura 3.6 Estructura Carpeta Listeners	32
Figura 3.7 Estructura Carpeta Notificaciones.....	33
Figura 3.8 Estructura Carpeta Task	34
Figura 3.9 Estructura Carpeta Utils	35
Figura 3.10 Estructura Carpeta Res	35
Figura 3.11 Estructura Carpeta Drawable	36
Figura 3.12 Estructura Carpeta Layout	37
Figura 3.13 Estructura Carpeta Menú	37
Figura 3.14 Estructura Carpeta Values.....	38
Figura 3.15 Estructura del Servicio Web.....	40
Figura 3.16 Vista Alfresco	44
Figura 3.17 Puesta en Marcha Alfresco	45
Figura 3.18 Estructura de la Base de Datos.....	46
Figura 3.19 Puesta en Marcha de Base de Datos.....	47
Figura 3.20 Vista Base de Datos con ToadForMySql.....	48

Figura 4.1	Habilitación Orígenes Desconocidos	50
Figura 4.2	Vista Tarjeta SD paquete Eupct.apk	50
Figura 4.3	Desinstalar Aplicación Eupct	51
Figura 4.4	Pantalla Bienvenida Aplicación	52
Figura 4.5	Pantalla Acceso	53
Figura 4.6	Pantalla Acceso con certificado	53
Figura 4.7	Selección de Certificado	54
Figura 4.8	Navegación por los Espacios Disponibles	55
Figura 4.9	Visualización de Ficheros	56
Figura 4.10	Visualización del Fichero de Firma Digital	57
Figura 4.11	Descarga de Ficheros a Tarjeta SD a través de la Aplicación.....	58
Figura 4.12	Descarga de Ficheros a Tarjeta SD a través de Google Docs	59
Figura 4.13	Notificaciones	60
Figura 4.14	Ventana de Información	61
Figura 4.15	Salir de la Aplicación	62
Figura 4.16	Crear una Carpeta.....	63
Figura 4.17	Subir un Fichero	64
Figura 4.18	Subir Y Firmar un Fichero	66
Figura 4.19	Borrado de Carpeta / Fichero	67
Figura 4.20	Renombrado de Carpeta / Fichero	69
Figura 4.21	Acceso a Base de Datos con ToadForMysql	70
Figura 4.22	Tabla Usuarios Base de Datos.....	71
Figura 4.23	Tabla de Usuarios con el Nuevo Usuario	72
Figura 4.24	Pantalla Acceso Alfresco	73
Figura 4.25	Administración Alfresco	73

Figura 4.26 Crear un Espacio	74
Figura 4.27 Añadir Nombre Nuevo Espacio	75
Figura 4.28 Listado Espacios Disponibles Actualizados	75
Figura 4.29 Ver Detalles de un Espacio	76
Figura 4.30Recuperar UUID de un Espacio	77
Figura 4.31 Vista Tabla Pertenencia	79
Figura 4.32 Vista en la Aplicación del Nuevo Espacio Creado	79

Índice de Cuadros

Cuadro 2.1 Firma XadesDetached.....	23
Cuadro 3.1 Cuadro Manifiesto de la Aplicación	39
Cuadro 3.2 URL Servicios Alfresco.....	43
Cuadro 3.3 Hibernate.cfg.xml	43
Cuadro 3.4 Ejecutar la Base de Datos	47
Cuadro 4.1 Añadir un Nuevo Usuario	71
Cuadro 4.2 URL Administrador Web Alfresco.....	72
Cuadro 4.3 URL del Espacio Buscado	77
Cuadro 4.4 Insertar una Pertenencia a un Espacio	78



Capítulo 1

Introducción

La tendencia de la sociedad actual a la hora de acceder a contenidos ha cambiado radicalmente en los últimos años a partir del nacimiento de teléfonos móviles de alta capacidad y dispositivos móviles como por ejemplos las Tablets. Es cada vez más raro ver a un usuario que para consultar un documento o información se espere a llegar a un ordenador de sobremesa, teniendo un teléfono móvil de última generación.

Por otro lado no podemos negar que en un entorno universitario existen continuas colaboraciones e intercambio de información digital (sustituyendo cada vez más al formato físico) entre la comunidad que la integra. Son ejemplos de ello el intercambio de conocimiento entre profesor y alumnos en la impartición de asignaturas, y los grupos de investigación con personal PDI.

1.1 Motivación

Es por ello que se desarrolla este proyecto cuya misión es intentar aunar lo comentado anteriormente para intentar satisfacer una necesidad que en la actualidad va en aumento.

Hay que entender que el modelo real de compartir información no es único ni universal, por ejemplo, un profesor de un departamento puede estar dando clase en una asignatura de un grado de Telecomunicaciones, otro de Economía y a la vez llevar dos líneas de investigación con diferentes compañeros de departamento. Por tanto observamos la necesidad de que nuestra aplicación distinga la posibilidad de ver diferentes espacios, cada uno con unos usuarios, seguramente, distintos.

Por otro lado, y siguiendo con el ejemplo, no tienen la misma responsabilidad sobre un espacio de una asignatura un profesor que los alumnos, por los que los permisos que ambas “entidades” tengan sobre él, deben ser distintos.

No debemos olvidarnos que cubrir el máximo de usuarios posibles debe ser un objetivo y en este aspecto no cabe duda que el sistema operativo que más mercado abarca



es Android. Es por ello que debemos usar su modelo de estructura y desarrollo para llevarlo a cabo en nuestro proyecto.

Por último debemos pensar que no todos los usuarios potenciales de la herramienta tienen el mismo conocimiento de las tecnologías de la información, por lo que se debe mostrar una interactividad visualmente agradable y funcionalmente sencilla.

1.2 Objetivos del Proyecto

Según lo expuesto, el objetivo final del proyecto no debe ser otro crear una aplicación móvil en Android en el cual el usuario pueda compartir y nutrirse de información documental en diferentes espacios colaborativos.

Para ello se deberá realizar además un servidor capaz de recibir las consultas y peticiones de dicha aplicación móvil y dar respuesta adecuada a la acción deseada. Dicho servidor será el contenedor de toda la documentación, por lo que para su almacenamiento deberemos utilizar aplicaciones especialistas en esta finalidad, como es por ejemplo Alfresco.

Una vez garantizado el acceso y manejo de la información por parte de la aplicación móvil, debemos intentar dar más versatilidad y fiabilidad a toda esta información guardada. Es por ello que aprovechándonos que Android permite el almacenamiento de certificados y claves, podemos dar la opción de garantizar la autoridad de acceso y de los documentos compartidos mediante firma digital válida.

1.3 Estructura de la memoria

La estructura que seguiremos en este documento es: En el [capítulo 2](#) realizaremos una introducción a las tecnologías usadas e intentar explicar por qué se han usado para esta aplicación. En el [capítulo 3](#) mostraremos la implementación y el diseño que contiene nuestra aplicación.

Una vez explicado todo esto, nos quedaría relatar cómo se usa la herramienta y qué se puede hacer con ella. Será en el [capítulo 4](#) con el manual de usuario.

Por último en el [capítulo 5](#) realizaremos nuestras conclusiones sobre el proyecto y qué posibles líneas de desarrollo futuras se podrían realizar. El [capítulo 6](#) está destinado a las referencias usadas para documentarnos.



Capítulo 2

Contexto / Tecnologías

Este capítulo tiene por finalidad realizar una introducción poco profunda sobre las múltiples aplicaciones y tecnologías que se han usado durante el proyecto. Su finalidad debe ser doble intentar que el usuario comprenda la motivación por las que se han usado ellas y no otras entre todas las disponibles en el mercado y que se familiarice y relacione diferentes conceptos comúnmente usados en la memoria, como por ejemplo, gestor documental, *MySQL*, servicios web, firma digital, *xades*, etc.

2.1 S.O. Android

Es muy posible que la mayoría de los lectores de esta memoria de proyecto tenga bastante claro qué es Android, qué diferencias tiene con otros sistemas operativos y por qué tiene tanto crecimiento. Pero vemos necesario realizar una introducción sobre qué es Android.

Así que a modo de presentación, pasamos a explicar qué es Android, por qué se dice que es un sistema operativo para dispositivos móviles libre, cuál es su historia y cómo es el desarrollo de aplicaciones en él.

2.1.1 ¿Qué es Android?

Android es un sistema operativo inicialmente pensado para teléfonos móviles, al igual que iOS, *Symbian* y *Blackberry OS*. Lo que lo hace diferente es que está basado en *Linux*, un núcleo de sistema operativo libre, gratuito y multiplataforma.

El sistema permite programar aplicaciones en una variación de Java llamada *Dalvik*. El sistema operativo proporciona todas las interfaces necesarias para desarrollar aplicaciones que accedan a las funciones del teléfono (como el GPS, las llamadas, la agenda, etc.) de una forma muy sencilla en un lenguaje de programación muy conocido como es Java.



Esta sencillez, junto a la existencia de herramientas de programación gratuitas, hace que una de las cosas más importantes de este sistema operativo sea la cantidad de aplicaciones disponibles, que extienden casi sin límites la experiencia del usuario.

2.1.2 Libertad

Una de las mejores características de este sistema operativo es que es completamente libre. Es decir, ni para programar en este sistema ni para incluirlo en un teléfono hay que pagar nada. Y esto lo hace muy popular entre fabricantes y desarrolladores, ya que los costes para lanzar un teléfono o una aplicación son muy bajos.

Cualquiera puede bajarse el código fuente, inspeccionarlo, compilarlo e incluso cambiarlo. Esto da una seguridad a los usuarios, ya que algo que es abierto permite detectar fallos más rápidamente. Y también a los fabricantes, pues pueden adaptar mejor el sistema operativo a los terminales.

2.1.3 Un poco de historia

Android era un sistema operativo para móviles prácticamente desconocido hasta que en 2005 Google lo compró. Hasta noviembre de 2007 sólo hubo rumores, pero en esa fecha se lanzó la *Open Handset Alliance*, que agrupaba a muchos fabricantes de teléfonos móviles, chipsets y Google y se proporcionó la primera versión de Android, junto con el SDK para que los programadores empezaran a crear sus aplicaciones para este sistema.

Aunque los inicios fueran un poco lentos, debido a que se lanzó antes el sistema operativo que el primer móvil, rápidamente se ha colocado como el sistema operativo de móviles más vendido del mundo, situación que se alcanzó en el último trimestre de 2010.

En febrero de 2011 se anunció la versión 3.0 de Android, llamada con nombre en clave Honeycomb, que está optimizado para tabletas en lugar de teléfonos móviles. Por tanto Android ha trascendido los teléfonos móviles para trascender a dispositivos más grandes.

Debemos destacar la versión 4.0 (Ice Cream Sandwich) debido a que traía como novedad una de las características necesarias para nuestro desarrollo. No es otra que añadir un repositorio de certificados donde el usuario puede guardar con seguridad su par de claves y su certificado digital.

A la entrega de esta memoria, la versión actual de Android es la 4.1 llamada JellyBean.



2.1.4 Desarrollo y Estructura de las Aplicaciones

Cuando desarrollamos una aplicación para Android a través de su SDK, nos encontramos que se crea una estructura genérica, independientemente de su tamaño o complejidad, que es la siguiente:

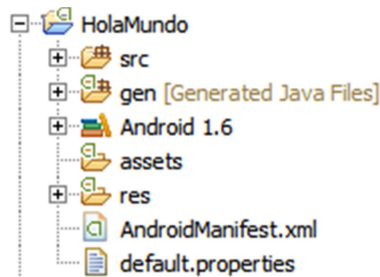


Figura 2.1 Estructura Proyecto Android

Describamos los elementos principales.

Carpeta /src/: Contiene todo el código fuente de la aplicación, código de la interfaz gráfica, clases auxiliares, etc.

Carpeta /res/: Contiene todos los ficheros de recursos necesarios para el proyecto: imágenes, vídeos, cadenas de texto, etc. Los diferentes tipos de recursos se deberán distribuir entre las siguientes carpetas:

- **/res/drawable/.** Contienen las imágenes de la aplicación. Se puede dividir en **/drawable-ldpi**, **/drawable-mdpi** y **/drawable-hdpi** para utilizar diferentes recursos dependiendo de la resolución del dispositivo.
- **/res/layout/.** Contienen los ficheros de definición de las diferentes pantallas de la interfaz gráfica. Se puede dividir en **/layout** y **/layout-land** para definir distintos layouts dependiendo de la orientación del dispositivo.
- **/res/anim/.** Contiene la definición de las animaciones utilizadas por la aplicación.
- **/res/menu/.** Contiene la definición de los menús de la aplicación.
- **/res/values/.** Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (**strings.xml**), estilos (**styles.xml**), colores (**colors.xml**), etc.
- **/res/xml/.** Contiene los ficheros XML utilizados por la aplicación.
- **/res/raw/.** Contiene recursos adicionales, normalmente en formato distinto a XML, que no se incluyan en el resto de carpetas de recursos.

Carpeta /gen/: Contiene una serie de elementos de código generados automáticamente al compilar el proyecto. Cada vez que generamos nuestro proyecto, la maquinaria de



compilación de Android genera por nosotros una serie de ficheros fuente en java dirigidos al control de los recursos de la aplicación.

Carpeta /assets/: Contiene todos los demás ficheros auxiliares necesarios para la aplicación (y que se incluirán en su propio paquete), como por ejemplo ficheros de configuración, de datos, etc.

Fichero AndroidManifest.xml: Contiene la definición en XML de los aspectos principales de la aplicación, como por ejemplo su identificación (nombre, versión, icono,...), sus componentes (pantallas, mensajes,...) o los permisos necesarios para su ejecución.

2.1.5 Componentes de una aplicación Android

En Android disponemos de componentes genéricos que suelen formar parte de una aplicación. Los detallamos brevemente:

Activity: Las *actividades* (activities) representan el componente principal de la interfaz gráfica de una aplicación Android. Se puede pensar en una *actividad* como el elemento análogo a una *ventana* en cualquier otro lenguaje visual.

View: Los objetos *view* son los componentes básicos con los que se construye la interfaz gráfica de la aplicación. De inicio, Android pone a nuestra disposición una gran cantidad de controles básicos, como cuadros de texto, botones, listas desplegables o imágenes, aunque también existe la posibilidad de extender la funcionalidad de estos controles básicos o crear nuestros propios controles personalizados.

Service: Los *servicios* son componentes sin interfaz gráfica que se ejecutan en segundo plano. En concepto, son exactamente iguales a los servicios presentes en cualquier otro sistema operativo. Los servicios pueden realizar cualquier tipo de acciones, por ejemplo actualizar datos, lanzar notificaciones, o incluso mostrar elementos visuales (*activities*) si se necesita en algún momento la interacción con el usuario.

Content Provider: Un contentprovider es el mecanismo que se ha definido en Android para compartir datos entre aplicaciones. Mediante estos componentes es posible compartir determinados datos de nuestra aplicación sin mostrar detalles sobre su almacenamiento interno, su estructura, o su implementación. De la misma forma, nuestra aplicación podrá acceder a los datos de otra a través de los contentprovider que se hayan definido.

Broadcast Receiver: Un broadcast receiver es un componente destinado a detectar y reaccionar ante determinados mensajes o eventos globales generados por el sistema (por ejemplo: “Batería baja”, “SMS recibido”, “Tarjeta SD insertada”, ...) o por otras aplicaciones (cualquier aplicación puede generar mensajes (intents, en terminología



Android) broadcast, es decir, no dirigidos a una aplicación concreta sino a cualquiera que quiera *escucharlo*).

Widget: Los *widgets* son elementos visuales, normalmente interactivos, que pueden mostrarse en la pantalla principal (*home screen*) del dispositivo Android y recibir actualizaciones periódicas. Permiten mostrar información de la aplicación al usuario directamente sobre la pantalla principal.

Intent: Un *intent* es el elemento básico de comunicación entre los distintos componentes Android que hemos descrito anteriormente. Se pueden entender como los *mensajes* o *peticiones* que son enviados entre los distintos componentes de una aplicación o entre distintas aplicaciones. Mediante un *intent* se puede mostrar una *actividad* desde cualquier otra, iniciar un servicio, enviar un mensaje *broadcast*, iniciar otra aplicación, etc.

2.2 Gestor Documental Alfresco

Alfresco es el sistema de gestión de contenidos empresariales de código abierto que está ampliamente extendido en el mercado mundial. Está basado en tecnologías libres lo que hace que contenga unas grandes ventajas respecto a sus rivales:

Versión Gratuita totalmente funcional: Dispone de una “versión de la comunidad” (CommunityEdition) que es gratuita. Este tipo de herramientas tienen sus principales ingresos en soporte, formación y funcionalidades avanzadas, por lo que su puesta en marcha no tiene por qué tener un coste inicial.

Código Abierto: Cualquiera puede modificar el código a su gusto para adaptarlo a sus necesidades.

2.2.1 Arquitectura

Una vez descargado e instalado en nuestro servidor, podemos ver que Alfresco viene con multitud de aplicaciones y tecnologías incluidas, todas de código libre como hemos comentado con anterioridad.

Además del repositorio de contenidos, que es el núcleo de un gestor documental, Alfresco nos ofrece un framework de portal web para administrar y usar contenido, una interfaz CFIS compatible con Windows y sistemas operativos Unix, un servidor Tomcat para su puesta en marcha, un motor Lucene para la indexación de búsquedas, una base de datos Postgres para almacenar información, así como otras características menos relevantes a nuestro juicio.



2.3 Firma Digital

Una firma digital es un esquema matemático que sirve para demostrar la autenticidad de un mensaje digital, que puede ser por ejemplo un documento electrónico. Una firma digital da al destinatario seguridad de que el mensaje fue creado por el remitente (autenticidad de origen), y que no fue alterado durante la transmisión (integridad). Las firmas digitales se utilizan comúnmente para la distribución de software, transacciones financieras y en otras áreas donde es importante detectar la falsificación y la manipulación.

La firma digital consiste en un método criptográfico que asocia la *identidad* de una persona o de un equipo informático, al mensaje o documento. En función del tipo de firma, puede además asegurar la integridad del documento o mensaje.

La firma electrónica, como la firma hológrafa (autógrafa, manuscrita), puede vincularse a un documento para identificar al autor, para señalar conformidad (o disconformidad) con el contenido, para indicar que se ha leído y, en su defecto mostrar el tipo de firma y garantizar que no se pueda modificar su contenido.

2.3.1 XAdES

En nuestra aplicación realizamos un tipo de firma basada en una estructura XAdES-BES (forma básica que cumple los requerimientos legales de firma digital).

XAdES es un conjunto de extensiones a las recomendaciones XML-DSig haciéndolas adecuadas para la firma electrónica avanzada.

Mientras que XML-DSig es un entorno general para firmar digitalmente documentos XML, XAdES especifica perfiles precisos de XML-DSig para ser usados con firma electrónica reconocida por la Unión Europea. Un beneficio importante de XAdES es que los documentos firmados electrónicamente pueden seguir siendo válidos durante largos períodos, incluso en el caso de que los algoritmos criptográficos subyacentes hayan sido rotos.

Existen 3 tipos principales de firma digital XAdES, siendo el tipo “Detached” el implementado por nosotros.

2.3.2 Estructura XAdESDetached

En este tipo de firmas, el elemento que se firma es un recurso externo que estará referenciado por una URI o referencia externa.



Una firma XAdES-BES Detached tiene el siguiente aspecto:

```
<?xml version="1.0" encoding="UTF-8"?>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:etsi="http://uri.etsi.org/01903/v1.3.2#" Id="Signature504735">
  <ds:SignedInfo Id="Signature-SignedInfo1024952">
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-
      20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference Id="SignedPropertiesID429729"
      Type="http://uri.etsi.org/01903#SignedProperties" URI="#Signature504735-
        SignedProperties48056">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue><!-- Digest del elemento referenciado en Base64 --></ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#Certificatel237555">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue><!-- Digest del elemento referenciado en Base64 --></ds:DigestValue>
    </ds:Reference>
    <ds:Reference Id="Reference-ID-200615" URI="file:/C:/Temp/ExampleToSign.xml">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue><!-- Digest del elemento referenciado en Base64 --></ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue Id="SignatureValue552465"><!-- Valor de la firma en Base64 -->
</ds:SignatureValue>
  <ds:KeyInfo Id="Certificatel237555">
    <ds:X509Data>
      <ds:X509Certificate><!-- Certificado firmante en Base64 -->
</ds:X509Certificate>
    </ds:X509Data>
    <ds:KeyValue>
      <ds:RSAKeyValue>
        <ds:Modulus><!-- Módulo de la clave RSA en Base64 --></ds:Modulus>
        <ds:Exponent><!-- Exponente de la clave RSA en Base64 --></ds:Exponent>
      </ds:RSAKeyValue>
    </ds:KeyValue>
  </ds:KeyInfo>
  <ds:Object Id="Signature504735-Object873466">
    <etsi:QualifyingProperties Target="#Signature504735">
      <etsi:SignedProperties Id="Signature504735-SignedProperties48056">
        <etsi:SignedSignatureProperties>
          <etsi:SigningTime><!-- Fecha y hora de la firma --></etsi:SigningTime>
          <etsi:SigningCertificate>
            <etsi:Cert>
              <etsi:CertDigest>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue><!-- Digest del certificado en Base64 --></ds:DigestValue>
              </etsi:CertDigest>
              <etsi:IssuerSerial>
                <ds:X509IssuerName><!-- Nombre de emisión del certificado firmante -->
</ds:X509IssuerName>
                <ds:X509SerialNumber><!-- Número de serie del certificado firmante -->
</ds:X509SerialNumber>
              </etsi:IssuerSerial>
            </etsi:Cert>
          </etsi:SigningCertificate>
        </etsi:SignedSignatureProperties>
        <etsi:SignedDataObjectProperties>
          <etsi:DataObjectFormatObjectReference="#Reference-ID-200615">
            <etsi:Description><!-- Descripción objeto firmado ---></etsi:Description>
            <etsi:MimeType><!-- Tipo MIME del objeto firmado ---></etsi:MimeType>
          </etsi:DataObjectFormat>
        </etsi:SignedDataObjectProperties>
      </etsi:SignedProperties>
    </etsi:QualifyingProperties>
  </ds:Object>
</ds:Signature>
```

Cuadro 2.1 Firma XadesDetached



2.4 Base de Datos MySql

MySQL es un sistema de gestión de bases de datos relacional, multihijo y multiusuario ampliamente utilizado en todo tipo de aplicaciones. Su gran fuerza es que es gratuito lo que hace que para desarrollos sin alta complejidad ni carga de transacciones sea una muy buena opción para reducir costes.

MySQL es una base de datos muy rápida en la lectura pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Por tanto nos ofrece una solución ideal para guardar la información de manejará el servidor de nuestra aplicación.

2.5 Servicios Web

Un servicio web (en inglés, *Web service*) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Algunas ventajas serían:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Utilizan normalmente protocolo HTTP sobre TCP en el puerto 80 lo que hace que en la mayoría de subredes empresariales o universitarias donde hay firewall por parte del administrador, no haya ningún problema en su utilización. Para algunos esta ventaja no es tal, sino que es una desventaja como comentaremos a continuación.



Algunos desventajas serían:

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación.
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI, CORBA o DCOM . Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en *firewall* cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

2.5.1 Protocolo SOAP

Ya que es el protocolo específico que hemos utilizado para implementar los servicios web de nuestra aplicación, vamos a definir brevemente qué es.

SOAP (siglas de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo basado en XML consiste de tres partes: un sobre (envelope), el cual define qué hay en el mensaje y cómo procesarlo; un conjunto de reglas de codificación para expresar instancias de tipos de datos; y una conversión para representar llamadas a procedimientos y respuestas.

2.6 Hibernate

Hibernate es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Sus características principales serían:

- Nos abstrae de la base de datos utilizada con una simple declaración en su fichero de configuración.
- Es software libre, y por tanto gratuito.
- Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.



- Hibernate ofrece también un lenguaje de consulta de datos llamado **HQL** (*HibernateQueryLanguage*), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "*criteria*").

2.7 Herramientas Utilizadas Desarrollo

A continuación vamos a detallar también brevemente algunos de los frameworks usados para el desarrollo del proyecto.

2.7.1 Eclipse + Android SDK

Ha sido el entorno principalmente usado para el desarrollo del código tanto del cliente como del servidor. Además para la parte de la aplicación del cliente, nos hemos apoyado en la librerías Android SDK ya que se integra perfectamente con Android.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE). Hemos obtenido por la versión de Eclipse Java EE Juno (Eclipse 4.2) que es la última en el momento de entra de este proyecto.

El SDK (SoftwareDevelopment Kit) de Android, incluye un conjunto de herramientas de desarrollo. Las plataformas de desarrollo soportadas incluyen Linux (cualquier distribución moderna), Mac OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo soportada oficialmente es Eclipse junto con el complemento ADT (AndroidDevelopment Tools plugin). Además, pueden controlarse dispositivos Android que estén conectados (reiniciarlos, instalar aplicaciones en remoto, etc.).

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo. Un paquete APK incluye ficheros .dex(ejecutablesDalvik, un código intermedio compilado), recursos, etc.

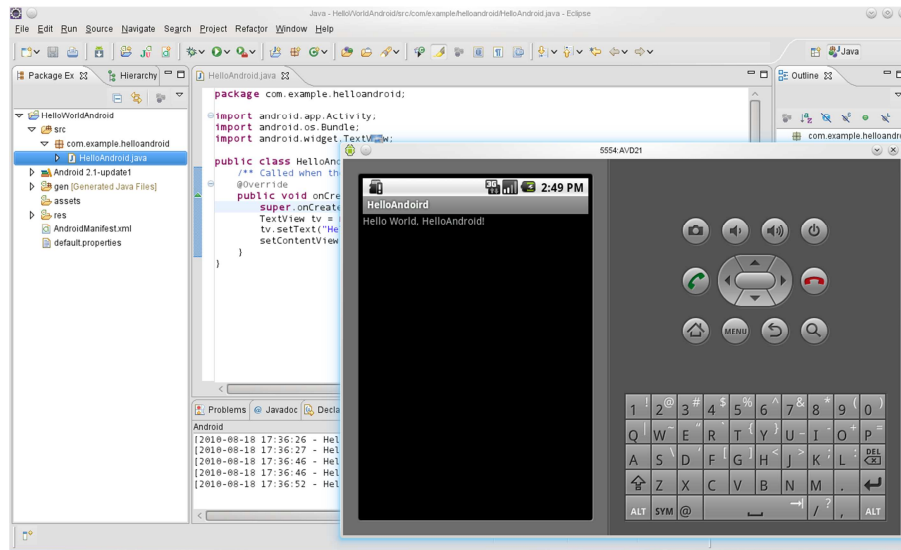


Figura 2.2 Eclipse + Android SDK

2.7.2 ToadforMySQL

TOAD es una aplicación informática de desarrollo SQL y administración de base de datos, considerada una herramienta útil para los Oracle DBAs (administradores de base de datos). Actualmente está disponible para las siguientes bases de datos: Oracle Database, Microsoft SQL Server, IBM DB2, y MySQL.

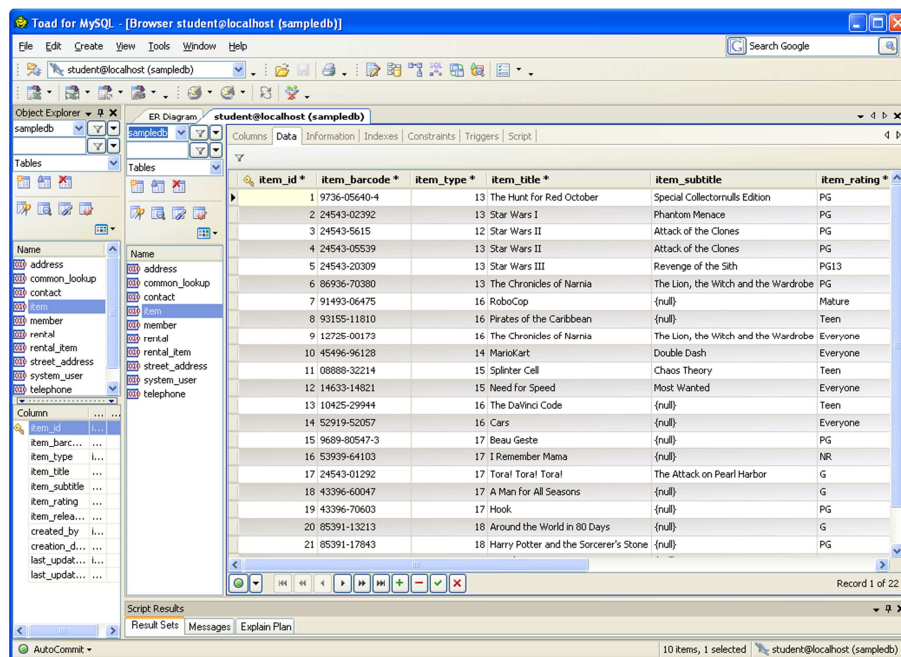


Figura 2.3 ToadForMySQL



Capítulo 3

Implementación de la Aplicación

En este capítulo explicaremos cómo hemos implementado toda la aplicación, tecnologías usadas y estructura realizada.

3.1 Objetivos de la implementación

A la hora de desarrollar todos los requisitos del proyecto, nos planteamos ciertos objetivos que debemos cumplir a la hora de llevarlo a cabo de tal forma que para posibles cambios de un desarrollador posterior o cambio en la arquitectura sea realizable sin tener que reescribir todo el código.

Algunos de los puntos principales de estas buenas prácticas son:

- **Interfaz simple y de fácil.** Que el usuario final no tenga grandes complicaciones en encontrar la acción que quiere realizar con un estilo agradable y dinámico.
- **Cambios en Arquitectura.** Es muy importante tener en cuenta que algunas de nuestras tecnologías puede cambiar con el paso del tiempo y por tanto la herramienta debe seguir funcionando. Un ejemplo muy claro es por ejemplo un cambio de la base de datos de MySQL a por ejemplo Oracle. Debido a la utilización en el servidor de servicios web de Hibernate para conectar con la base de datos, nos sirve de conector “transparente” ante un posible cambio de ese tipo, por lo que sería fácil realizarlo.
- **Uso de Estándares.** Otra cualidad importante es usar lo máximo posible estándares de tecnología para que haya suficiente conocimiento y posibilidad de compatibilidad con otras herramientas. En esta aplicación hemos utilizado multitud de estándares: Servicios Web de primera generación SOAP, estructura de fichero de firma digital XADES-BES, etc.



- **Código fácilmente modificable.** Debido a que este proyecto tiene múltiples posibilidades de desarrollo futuras (como se indica en el apartado [Líneas Futuras](#)) es muy importante que el código tenga una estructura sencilla y esté bien comentado.
- **Aplicación ligera para Android.** Ya que los recursos de los teléfonos móviles suelen ser más limitados que los de los ordenadores de sobremesa, hay que prestar atención en no crear una herramienta que ocupe mucho, que procese mucho y que por tanto sea muy pesada. Esto lo hemos tenido en cuenta a la hora de la firma digital, encargándose de las librerías pesadas el servidor y realizar una simple firma de un hash en el cliente.

3.2 Arquitectura del Proyecto

El proyecto distingue principalmente dos partes bien diferenciadas, por un lado la aplicación del cliente, instalada en el móvil y que es la parte visual. Por otro lado el servidor, que contiene todo el núcleo de los servicios, la base de datos MySQL y el gestor documental Alfresco. La siguiente ilustración muestra esta arquitectura a alto nivel:

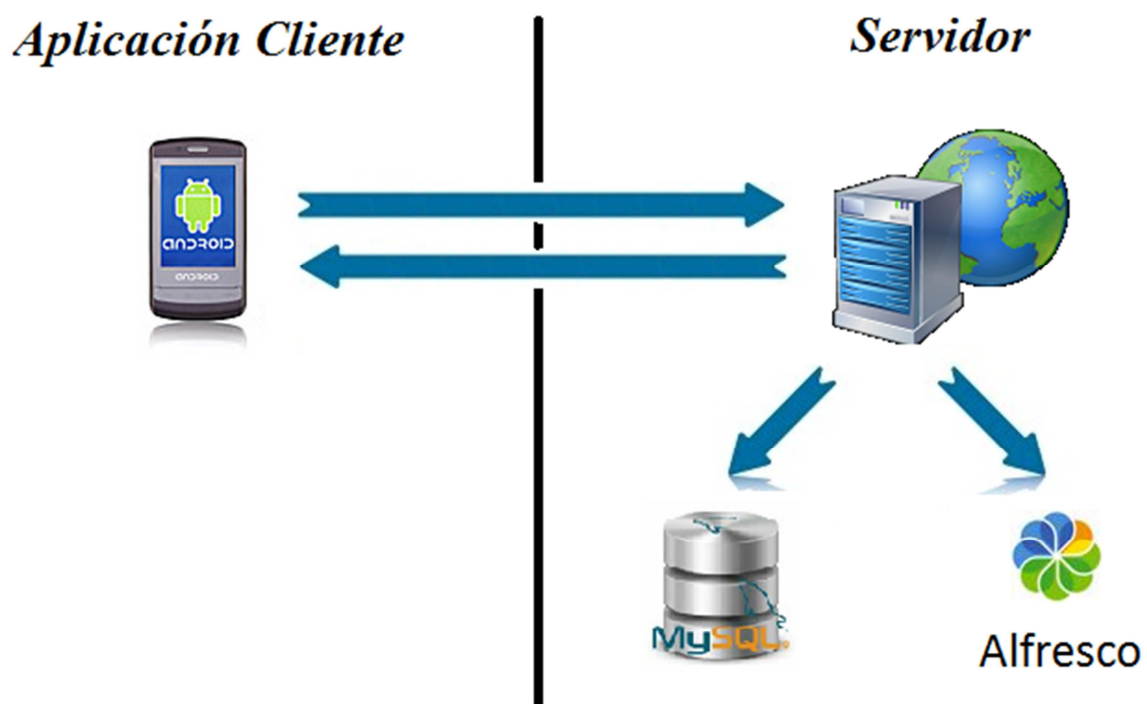


Figura 3.1 Arquitectura Proyecto



3.3 Aplicación Cliente

En este apartado vamos a explicar todo el desarrollo e implementación de la aplicación del cliente (la instalada en el sistema operativo Android).

3.3.1 Estructura de la Aplicación

El esqueleto seguido para el desarrollo de esta parte de la aplicación es la estándar de toda aplicación para Android y la que nos autogenera inicialmente el plugin SDK de Android para Eclipse.

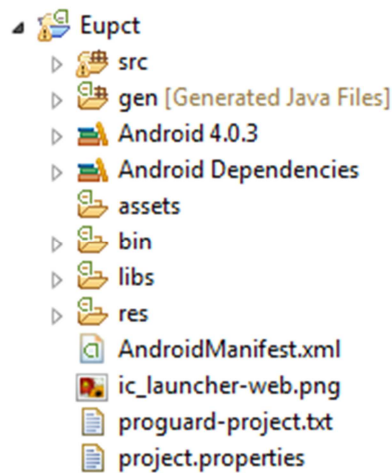


Figura 3.2 Estructura Aplicación Cliente

A continuación vamos a entrar un poco en detalle sobre las clases, sus funciones y recursos del proyecto.

3.3.2 Implementación Clases - Directorio src

En este directorio se encuentran los paquetes Java y el código fuente de las clases que componen la aplicación. Vamos a ir comentando cada uno de estos paquetes, clases destacables y resaltar la funcionalidad más destacable.

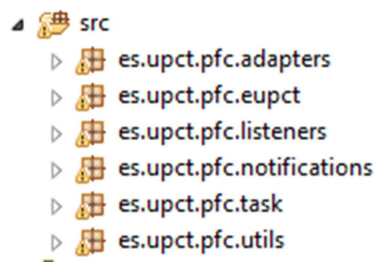


Figura 3.3 Estructura Carpeta Src

3.3.2.1 es.upct.pfc.adapters

En este paquete tenemos las clases de adaptadores de Android para mostrar un listado de componentes.

CustomAdapter: Extiende del adaptador base de Android y lo que realiza es ir creando cada fila del listado para la navegación por carpetas. Según sea carpeta o fichero le va añadiendo la imagen y el texto correspondiente. También se encarga de añadir a cada fila las referencias hacia los “Listeners” para ejecutar acciones al hacer click sobre un elemento.

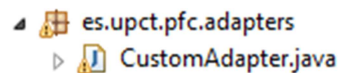


Figura 3.4 Estructura Carpeta Adapters

3.3.2.2 es.upct.pfc.eupct

Este paquete contiene las clases principales de ejecución en Android y son principalmente extensiones de “Activity”.

IntroActivity: Es la primera actividad que se ejecuta al lanzar la aplicación. Su función principal es mostrar la imagen de la UPCT de bienvenida y redirigir a la actividad de Login.

Login: Es la actividad donde el usuario introduce sus credenciales para acceder a la parte restringida de la aplicación. No sólo maneja el acceso por usuario y contraseña, sino que además muestra el acceso por certificado instalado en el repositorio de certificados del sistema operativo.



HomePage: Es la actividad que maneja toda la interacción en la navegación por los distintos espacios colaborativos. Es la clase más extensa y con más funcionalidad, ya que tiene que orquestar todas las acciones y llamar al resto de clases correspondientes para ejecutarlas.

Además para la subida o bajada de nivel en la navegación o recarga del explorador, se ejecuta dicha actividad de nuevo, realizando por tanto un ahorro considerable de recursos y clases extras.

Entre otras funcionalidades destacables, es la clase que se encarga de lanzar la búsqueda de notificaciones periódicamente.

ShowFile: Es la actividad en la cual insertamos un visor web para la visualización de documentos en Google Docs.

MyApp: No es una actividad, como los casos anteriores, sino que es una extensión de “Application” y lo que realiza es definir variables generales a toda la aplicación para que puedan ser llamadas por todo el resto de actividades. Aquí es donde se definen, entre otras cosas, la URL y nombre de los servicios web del servidor.

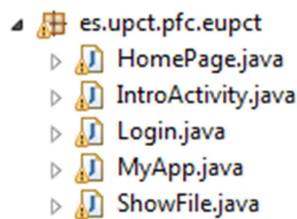


Figura 3.5 Estructura Carpeta Principal

3.3.2.3 es.upct.pfc.listeners

Paquete que contiene una única clase con las acciones que realizan los escuchadores:

Listeners: Define las acciones al pulsar (click) y mantener pulsado (longclick) tanto para carpetas como ficheros. Además estas acciones y los menús contextuales generados son variables según los permisos del usuario activo sobre ese recurso.

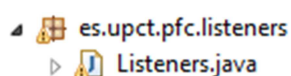


Figura 3.6 Estructura Carpeta Listeners



3.3.2.4 es.upct.pfc.notifications

Definimos aquí las clases referentes al manejo de notificaciones.

NewDocumentChecker: Es nuestra única clase de manejo de notificaciones. Este código se ejecuta periódicamente y según la clase HomePage tiene definido. Su funcionalidad es llamar al servidor y que le indique si el usuario actual tiene notificaciones pendientes. Si así fuera, se lo muestra a modo de aviso emergente en Android.

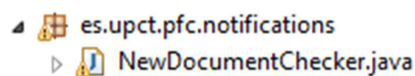


Figura 3.7 Estructura Carpeta Notifications

3.3.2.5 es.upct.pfc.task

Este paquete contiene las tareas asíncronas que ejecutamos contra el servidor. Todas tienen una estructura común por lo que el código es bastante similar: Recoger las variables de entrada, llamada al servicio web correspondiente, tratamiento de la respuesta y reenvío a la actividad correspondiente.

CallWSChildrenTask: Realiza el listado de las carpetas y ficheros de la carpeta seleccionada ya sea de forma ascendente o descendente

CallWSCreateSpaceTask: Tarea que se encarga de la creación de carpetas en el gestor documental a través del servicio web.

CallWSDeleteNodeTask: Tarea que realiza el borrado tanto de ficheros como de carpetas.

CallWSGetFileTask: Obtiene del Servidor la url para visualización de un fichero que se ha seleccionado.

CallWSGetFileToSaveTask: A diferencia de la tarea anterior, aquí obtenemos directamente del servidor el documento entero, de tal forma que podremos guardarlo en la tarjeta SD.

CallWSLoginTask: Maneja la comprobación de los credenciales tanto para usuario y contraseña como para certificado digital.

CallWSNotificacionesTask: Tarea que realiza la búsqueda de las nuevas notificaciones del usuario.



CallWSRenameNodeTask: Realiza el renombrado tanto de carpetas como de los ficheros que hay en el gestor documental.

CallWSRolesTask: Su función es recoger los permisos del usuario de la carpeta que está visualizando en ese momento.

CallWSUploadFileTask: Tarea de subida de ficheros a la carpeta actual.

CallWSUploadFileSignTask: Además de realizar la función de subida de documento, recibe y envía a la actividad el hash correspondiente para firmar. Posteriormente realiza la subida de éste hash firmado, asociándolo también al fichero original.

En la versión final no es utilizada ya que es sustituida por un desarrollo avanzado en la aplicación, que es la tarea siguiente.

CallWSUploadFileSignXadesTask: Tarea que realiza la función de subida de documento. Además, maneja el hash de firma y, una vez firmado, lo reenvía al servidor. La diferencia principal respecto a la versión “simple” de firma (la tarea *CallWSUploadFileSignTask*) es que aquí se envía al servidor el certificado del usuario para la realización de la estructura XADES del fichero XML de firma.

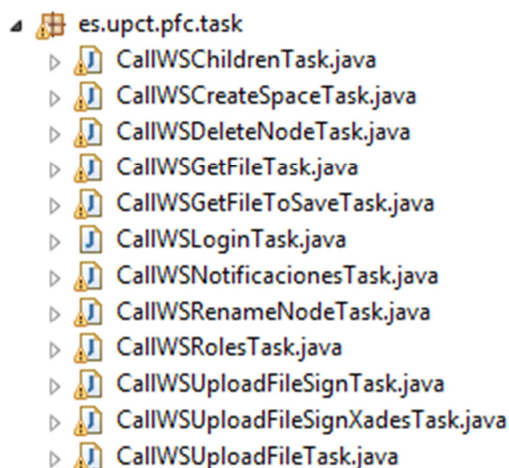


Figura 3.8 Estructura Carpeta Task

3.3.2.6 es.upct.pfc.utils

Este paquete contiene clases auxiliares que apoyan la funcionalidad del resto de clases. Son clases independientes y normalmente métodos que pueden ser usados desde diferentes actividades.



FileExplore: Clase que genera el listado de ficheros de la tarjeta SD y que devuelve el fichero seleccionado.

Utils: Clase general de contiene métodos generales que pueden ser referenciados desde cualquier otro método. Destacan métodos de conversión, el método de firma del hash junto con la clave privada, métodos de codificación y decodificación en Base64...

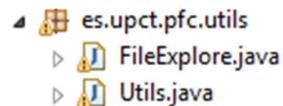


Figura 3.9 Estructura Carpeta Utils

3.3.3 Implementación de Interfaz - Directorio res

Directorio donde se almacenan los recursos externos de la aplicación. Estos recursos suelen ser las imágenes, parte visual, menús contextuales, ficheros de idioma, etc.

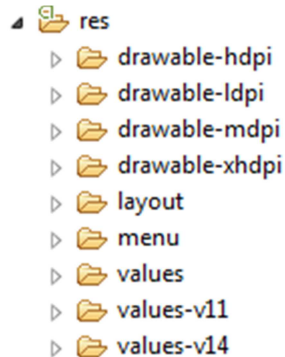


Figura 3.10 Estructura Carpeta Res

3.3.3.1 Drawable

Carpeta donde se almacenan todas las imágenes usadas por la aplicación. Se distinguen 4 categorías de imágenes :hdpi (imágenes pesadas), ldpi (imágenes ligeras), mdpi (imágenes de densidad normal) y xhdpi (imágenes muy pesadas).

En su mayoría, nosotros tenemos iconos por lo que estarán casi todos en la carpeta drawable-ldpi. Dejaremos las imágenes de cabecera y logo de la Upct como drawable-xdpi.

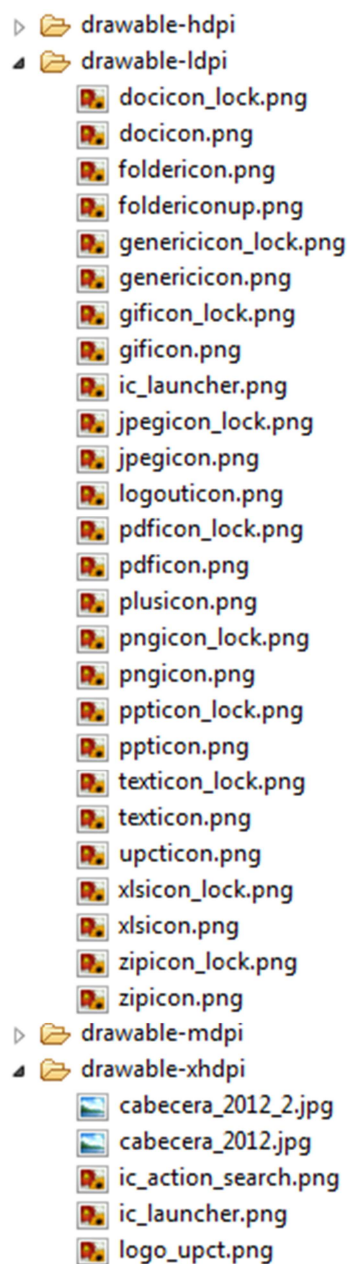


Figura 3.11 Estructura Carpeta Drawable

3.3.3.2 Layout

En esta carpeta definimos toda la parte visual de las diferentes actividades: disposición de imágenes, texto, colores, etc.

Activity_login: Contiene el layout de la ventana de logeo.

Activity_pag_principal: Contiene el layout de la ventana de exploración de carpetas y ficheros.



Activity_pag_principal_row: Está relacionado con la distribución anterior y contiene la estructura de cada fila del explorador de carpetas y ficheros. En nuestro caso tendremos una imagen (la correspondiente con el tipo del elemento) y un texto con su nombre.

Activity_show_file: Parte visual donde mostramos un fichero online, embebiendo en su interior la llamada a Google Docs.

Activity_splash_app: Es la pantalla de bienvenida.

Custom_rename: Diálogo que mostramos cuando queremos renombrar una carpeta o fichero. También se ha reutilizado para que sea también válido a la hora de crear una nueva carpeta.

Fileexplorer: Apariencia visual del listado de ficheros y carpetas de la tarjeta SD utilizado para la subida de documentos.

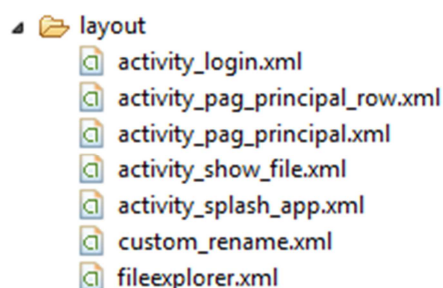


Figura 3.12 Estructura Carpeta Layout

3.3.3.3 Menu

En esta carpeta definimos los menús contextuales que tienen nuestras actividades cuando el usuario pulsa la tecla “MENÚ” de su teléfono móvil.

Activity_pag_principal_member: Es el menú que se muestra cuando el usuario tiene permisos de miembro (invitado) sobre un espacio de colaboración.

Activity_pag_principal: Es el menú que se le muestra al administrador (o gestor) de un espacio colaborativo. Evidentemente este menú tiene todas las opciones del menú de miembros más algunas funciones avanzadas.

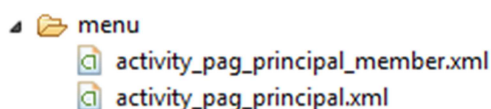


Figura 3.13 Estructura Carpeta Menu



3.3.3.4 Values

Carpeta donde se almacena valores dinámicos de recursos necesarios por la aplicación. Normalmente se destina a recursos de tipo idioma para mostrar el texto de las pantallas que ve el usuario dinámicamente, pudiendo por tanto ser cambiado en tiempo de ejecución ante un cambio de idioma, por ejemplo.

Firma: Fichero que define una estructura de recurso. La usamos para decir que un fichero firmado tiene una estructura de recurso compuesta por su documento original más el XML de firma asociado.

Strings: Contiene las cadenas de texto usadas por la aplicación. Todos los textos de los componentes de los diversos layouts deberían estar ahí, aunque luego en una actividad puedan ser cambiados dinámicamente.

Styles: Fichero predefinido que lo que realiza es aplicar una apariencia general a la aplicación.

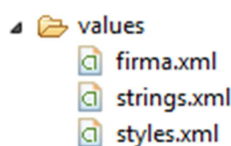


Figura 3.14 Estructura Carpeta Values

3.3.4 Manifiesto de la aplicación

Este documento es realmente importante en la aplicación, ya que define todos los recursos usados así como los permisos que necesita en el sistema operativo para su correcto funcionamiento. Además, también se define en él la versión de Android que requiere esta aplicación para ser 100% compatible.

A continuación lo añadimos. Podemos observar que se pide una versión del Sdk de Android 15, equivalente a Android 4.0. Esto es debido a que el repositorio de certificados propio no se instauró en Android hasta esa versión. Para todo lo demás, debería ser compatible con versiones 2.3 y posteriores. Además nuestra aplicación sólo tiene definidos dos permisos: acceso a internet para llamar al servicio web, acceder a Google Docs, etc y acceso de escritura a la tarjeta SD, para poder descargar documentos en ella.



```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="es.upct.pfc.eupct"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="15" />
    <uses-permission android:name="android.permission.INTERNET">
    </uses-permission>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" android:name=".MyApp">
        <activity android:name=".IntroActivity" android:label="@string/title_activity_main"
            android:theme="@android:style/Theme.NoTitleBar" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Login" android:label="@string/title_activity_login"
            android:theme="@android:style/Theme.NoTitleBar" >
            <meta-data
                android:name="android.support.PARENT_ACTIVITY"
                android:value="es.upct.pfc.eupct.SplashAppActivity" />
            </activity>
            <activity
                android:name=".HomePage" android:label="@string/title_activity_pag_principal"
                android:theme="@android:style/Theme.NoTitleBar" >
                <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
            <activity
                android:name=".ShowFile" android:label="@string/title_activity_show_file"
                android:theme="@android:style/Theme.NoTitleBar">
                <intent-filter>
                    <action android:name="android.intent.action.MAIN" />
                    <category android:name="android.intent.category.LAUNCHER" />
                </intent-filter>
            </activity>
            <service android:name="es.upct.pfc.notifications.NewDocumentChecker" />
        </application>
    </manifest>

```

Cuadro 3.1 Cuadro Manifiesto de la Aplicación

3.4 Servidor

En este apartado vamos a intentar explicar todas las tecnologías utilizadas así como la implementación de las diversas partes y su desarrollo. Para la aplicación cliente esta parte se comporta como un todo en uno, totalmente oculta para ella, salvo los servicios web.

3.4.1 Servicios Web



Los servicios web es el otro gran desarrollo de código realizado en el proyecto. Es la única parte visible entre Cliente - Servidor y sirve por tanto de máscara para la conexión con la base de datos y el gestor documental donde almacenamos los ficheros.

3.4.1.1 Estructura del Servidor de Servicios Web

El Servidor de Servicios Web, llamado en el código WSEUpct, no es más que un número de métodos que se pueden llamar remotamente. Decir que este componente lo hemos lanzado para su funcionamiento en un servidor de aplicaciones Tomcat v.6.0. (Puerto 8888), pero que puede ser desplegado en otros como por ejemplo Jboss.

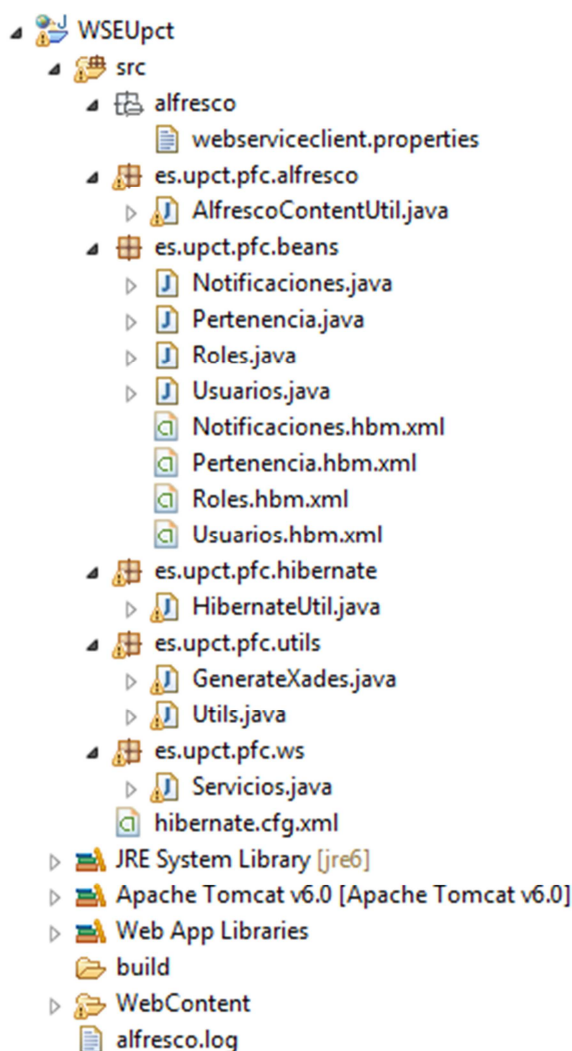


Figura 3.15 Estructura del Servicio Web

A continuación vamos a entrar un poco en detalle sobre las clases, paquetes, sus funciones y recursos del proyecto.



3.4.1.2 es.upct.pfc.ws

Definimos aquí la clase que sirve de EndPoint del Servicio Web.

Servicios: Es la clase más importante del proyecto, ya que se definen los métodos que son visibles desde el exterior del servidor, así como la funcionalidad que tienen y lo que devuelve cada uno. En él se pueden encontrar todos los métodos llamados por la aplicación cliente, como por ejemplo:

subir fichero (uploadFile),
borrar fichero (deletebyuuid),
comprobar credenciales(getName),
buscar notificaciones(getNotificaciones)

Podemos destacar el único método que contiene esta clase y que no es accesible desde el exterior, que es setNotificaciones. La razón por la que no es accesible es porque la aplicación en android no añade notificaciones, sino que éstas son automáticamente registradas en la base de datos cuando el servidor guarda un fichero en el gestor documental, siendo por tanto un método privado.

3.4.1.3 es.upct.pfc.alfresco

Paquete que contiene clases con funcionalidad particular del gestor documental Alfresco. Son las clases y métodos que realizan realmente la interacción con él.

AlfrescoContentUtil: Clase general con multitud de métodos utilizados para comunicarse con Alfresco. Son funcionalidades como recoger un recurso de gestor documental (recurso = nodo), borrar un nodo, renombrar un nodo, subir un documento, listar los nodos hijos de un nodo, etc.

3.4.1.4 es.upct.pfc.hibernate

Paquete que contiene las clases para interaccionar con Hibernate y por tanto con la base de datos MySql en este caso.



HibernateUtil: Clase general que maneja la conexión con la base de datos abriendo y cerrando las transacciones de ejecución de las sentencias SQL y cargando el fichero general de configuración *hibernate.cfg.xml* que explicaremos más adelante.

3.4.1.5 **es.upct.pfc.beans**

En este paquete es donde tenemos los beans o modelos de mapeo de tablas de la base de datos para pasarlo a objetos manejables en Java. Esta estructura es heredada del uso de hibernate.

Notificaciones y su fichero de configuración Notificaciones.hbm.xml: Mapean la tabla “Notificaciones” de la base de datos, teniendo tantas variables como columnas tienen dicha tabla en base de datos.

Pertenencia y su fichero de configuración Pertenencia.hbm.xml: Mapea la tabla “Pertenencia” de la base de datos.

Roles y su fichero de configuración Roles.hbm.xml: Mapea la tabla “Roles” de la base de datos.

Usuarios y su fichero de configuración Usuarios.hbm.xml: Mapea la tabla “Usuarios” de la base de datos

Por tanto lo que realmente estamos obteniendo son objetos que representan los registros de esas tablas en la base de datos, siendo así transparente a cualquier tipo o modelo de base de datos usada.

3.4.1.6 **es.upct.pfc.utils**

Paquete donde recogemos clases generales que puedan ser usadas a modo de utilidad por cualquier otra clase del proyecto.

GenerateXades: Clase donde su única función es generar toda la estructura de un fichero Xades-BES tipo “Detached” y rellenarlo con los datos del Certificado y el documento a firmar que le llegan.

Utils: Clase general con múltiples métodos de utilidades. Algunos comunes son conversores de un tipo de dato Java a otro, codificadores y decodificadores en Base64, devolver el mimetype de un fichero, etc.



3.4.1.7 alfresco

Paquete con un fichero de propiedades “webserviceclient.properties” en el cual sólo se indica la ubicación de la instalación de Alfresco. En nuestro caso:

```
repository.location=http://79.109.182.20:7777/alfresco/api
```

Cuadro 3.2 URL Servicios Alfresco

3.4.1.8 Hibernate.cfg.xml

Fichero de propiedades de Hibernate. En él vamos a destacar algunos parámetros usados como la url donde se encuentra el driver de la base de datos, su url, el esquema a usar, el usuario y la contraseña.

Las otras propiedades importantes es donde se definen los ficheros hbm de los beans, que sirven para mapear las tablas y columnas de la base de datos.

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<!-- Database connection settings -->
<property name="connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="connection.url">jdbc:mysql://localhost/servidor</property>
<property name="connection.username">root</property>
<property name="connection.password"></property>

<!-- JDBC connection pool (use the built-in) -->
<property name="connection.pool_size">1</property>
<!-- SQL dialect -->
<property name="dialect">org.hibernate.dialect.MySQLDialect</property>
<!-- Enable Hibernate's automatic session context management -->
<property name="current_session_context_class">thread</property>
<!-- Disable the second-level cache -->
<property
name="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>
<!-- Echo all executed SQL to stdout -->
<property name="show_sql">true</property>

<!-- Drop and re-create the database schema on startup -->
<property name="hbm2ddl.auto">update</property>

<mapping resource="es/upct/pfc/beans/Usuarios.hbm.xml" />
<mapping resource="es/upct/pfc/beans/Roles.hbm.xml" />
<mappingresource="es/upct/pfc/beans/Pertenencia.hbm.xml" />
<mappingresource="es/upct/pfc/beans/Notificaciones.hbm.xml" />
</session-factory>
</hibernate-configuration>
```

Cuadro 3.3 Hibernate.cfg.xml



Por tanto, si por ejemplo algún día necesitamos cambiar de base de datos, sólo deberemos cambiar el driver de conexión, la url nueva y los usuarios y contraseña si han cambiado. Todo lo demás es totalmente transparente para el Servicio Web.

3.4.2 Alfresco

El gestor documental utilizado, Alfresco, tiene una puesta en marcha bastante sencilla y la configuración realizada por nosotros es prácticamente nula.

3.4.2.1 Estructura del Gestor Documental

La estructura de nuestra instalación es bastante sencilla. No se ha configurado casi nada de forma avanzada, ya que para almacenar nuestros documentos era suficiente con lo que nos daba por defecto.

Comentar que Alfresco en su instalación inicial viene con una base de datos Postgres. Podíamos haber cambiado fácilmente a que conectara a otro esquema de nuestra base de datos MySQL, pero no hemos querido configurar más allá de lo suficiente y esa base de datos ya cubría ampliamente nuestras necesidades.

En cuanto al código de la aplicación, estará corriendo en un servidor de aplicaciones Tomcat v6.0 al que le hemos cambiado el puerto http a 7777. Este tipo de gestor documental tiene un acceso web bastante intuitivo y por tanto bastante fácil de administrar.

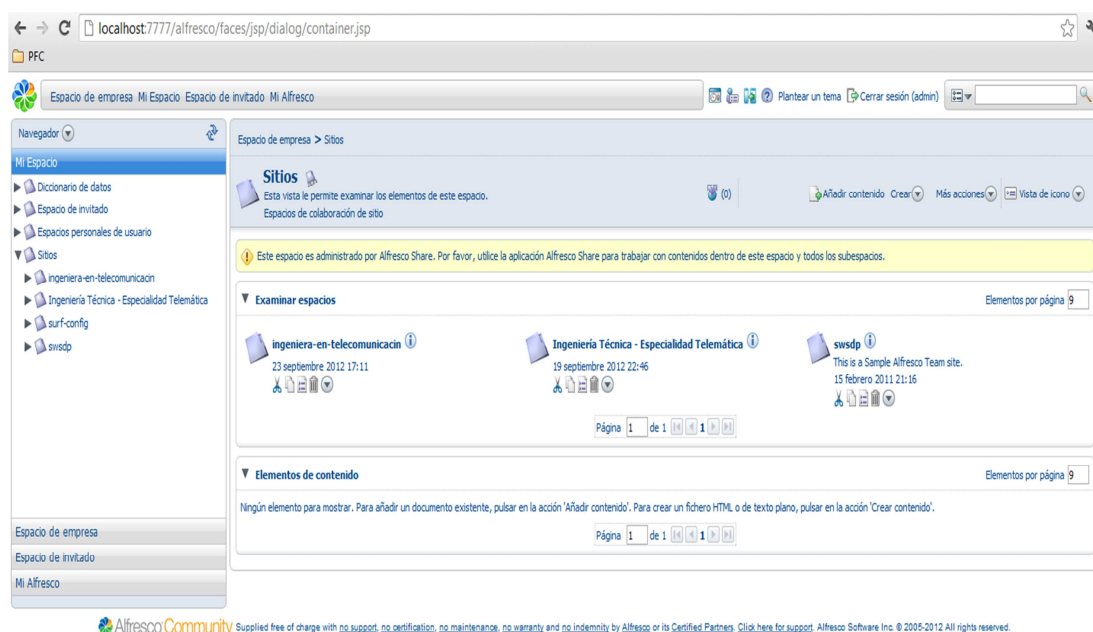


Figura 3.16 Vista Alfresco



3.4.2.2 Puesta en marcha

En Windows, la puesta en marcha es muy sencilla, ya que la versión actual de Alfresco te da una ventana de administración para arrancar o parar los servicios.

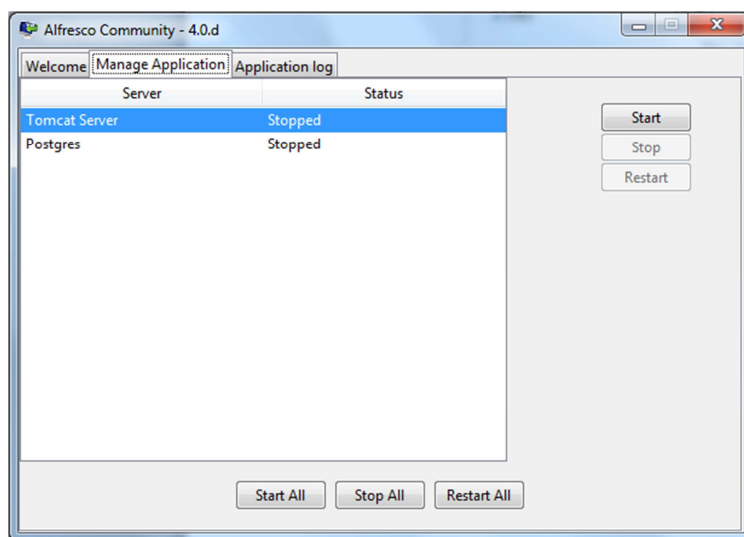


Figura 3.17 Puesta en Marcha Alfresco

Es recomendable primero lanzar la base de datos seleccionándola y pulsando el botón “Start” y luego lanzar el servidor Tomcat con el código del gestor documental.

Una vez nos indique que ambos servicios están “Running”, ya podremos trabajar con normalidad. Comentar por último algunos datos extras de configuración:

- Tomcat Server – Alfresco: Puerto 7777, usuario : admin, contraseña : admin
- Postgres: Puerto 5432, usuario : alfresco, contraseña : (no tiene). Se puede administrar con una herramienta como “pgAdmin III”.

3.4.3 MySql

La base de datos elegida para el Servidor es MySql debido a su amplia generalización de uso y sobre todo a que es gratuita. Al principio deberemos añadir nuestras tablas con sus diferentes columnas así como rellenar los datos adecuadamente a lo que necesitemos. Veamos los detalles.



3.4.3.1 Estructura de la Base de Datos

Nuestra base de datos tendrá solamente 4 tablas, que vienen dadas por el siguiente esquema Entidad – Relación:

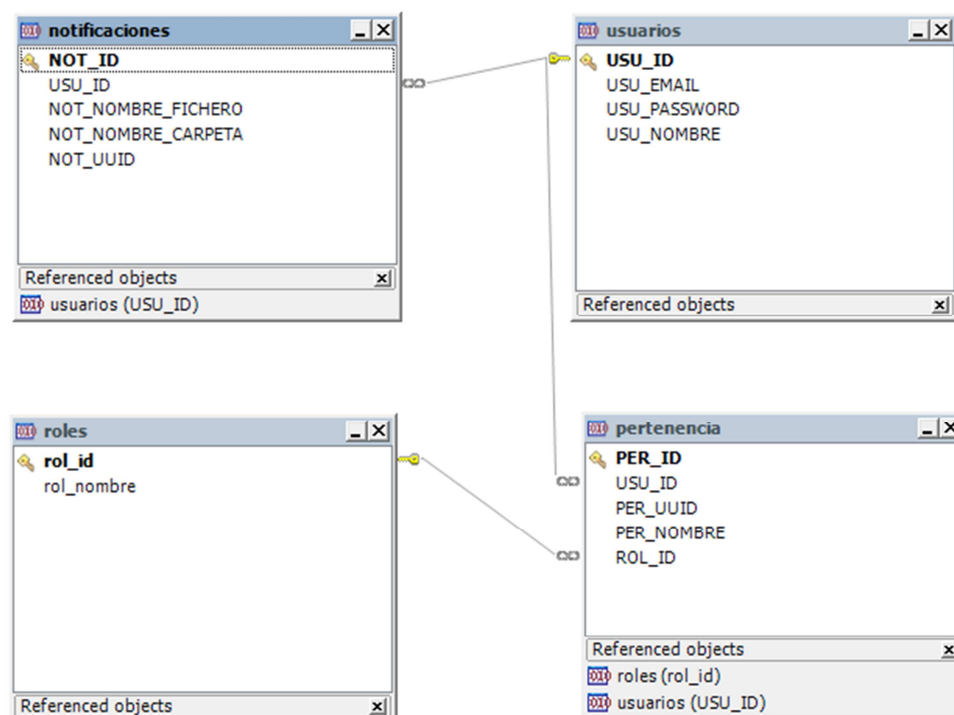


Figura 3.18 Estructura de la Base de Datos

A continuación vamos a explicar para qué sirve cada tabla :

Usuarios: Es la tabla donde daremos de alta nuestros usuarios en la plataforma. Será el identificador único su NIF o DNI.

Roles: Tabla donde indicamos los tipos de roles existentes en nuestra aplicación. Sólo tenemos 2 definidos: 1 – administrador, 2 – miembro.

Pertenencia: Relacionamos qué usuario tiene qué permisos sobre una carpeta (y sus subcarpetas). Estas carpetas vendrán dadas por los UUIDs únicos que nos proporciona Alfresco.

Notificaciones: Aquí están registradas las nuevas notificaciones que quedan pendientes por informar a los usuarios.



3.4.3.2 Puesta en marcha

Debido a que nos hemos descargado una versión de MySQL no instalable, debemos lanzarlo de forma manual con los comandos:

```
cd %MySQL_HOME%\bin  
mysqld --console
```

Cuadro 3.4 Ejecutar la Base de Datos

Una vez lanzado, nos aparecerá una ventana como ésta:

```
C:\Windows\system32\cmd.exe  
C:\Users\CASA\Desktop\PFC\Desarrollo>cd C:\Users\CASA\Desktop\PFC\Desarrollo\mysql\bin  
C:\Users\CASA\Desktop\PFC\Desarrollo\mysql\bin>mysqld --console  
120924 1:32:38 [Note] Plugin 'FEDERATED' is disabled.  
120924 1:32:38 InnoDB: The InnoDB memory heap is disabled  
120924 1:32:38 InnoDB: Mutexes and rw_locks use Windows interlocked functions  
120924 1:32:38 InnoDB: Compressed tables use zlib 1.2.3  
120924 1:32:38 InnoDB: Initializing buffer pool, size = 128.0M  
120924 1:32:38 InnoDB: Completed initialization of buffer pool  
120924 1:32:38 InnoDB: highest supported file format is Barracuda.  
120924 1:32:39 InnoDB: Waiting for the background threads to start  
120924 1:32:40 InnoDB: 1.1.5 started; log sequence number 7748703  
120924 1:32:43 [Note] Event Scheduler: Loaded 0 events  
120924 1:32:43 [Note] mysqld: ready for connections.  
Version: '5.5.10' socket: '' port: 3306 MySQL Community Server <GPL>
```

Figura 3.19 Puesta en Marcha de Base de Datos

Podemos observar que el puerto que está utilizando es el 3306. Podemos ya acceder a ella a través de cualquier herramienta de gestión de base de datos tipo Mysql. Los credenciales son:

User :root, Contraseña : (vacía)

En nuestro caso hemos usado ToadForMySQL, de tal forma que la apariencia que tiene es la siguiente.

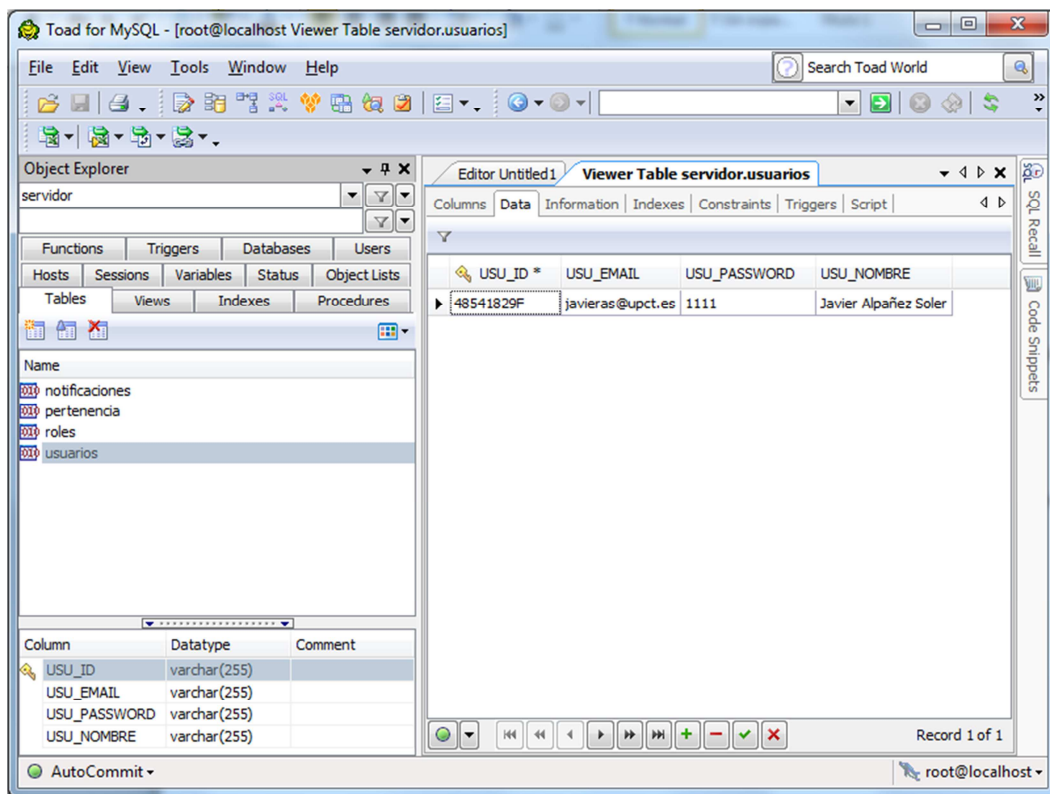


Figura 3.20 Vista Base de Datos con MySQL



Capítulo 4

Manual de Usuario

En este capítulo abarcaremos, inicialmente, desde el punto de vista del usuario todos los pasos a seguir para la correcta puesta en marcha de la aplicación así como casos de uso reales, todo ello acompañado de imágenes capturadas desde el móvil.

Posteriormente explicaremos cómo se administra la parte del servidor, cuyo esfuerzo estará centrado en la introducción de datos en la base de datos y el gestor documental.

4.1 Aplicación Cliente

Vamos a empezar definiendo cómo puede un usuario instalar la aplicación en su teléfono Android y qué se va a encontrar una vez lance la aplicación. Detallaremos ejemplos de uso de forma fácil e intuitiva acompañándolo de capturas reales.

4.1.1 Instalación / Desinstalación de la Aplicación

La instalación de aplicaciones en Android viene acometida principalmente por la aplicación Google Play que contiene Android. Desde ahí se pueden descargar e instalar la mayoría de aplicaciones disponibles desde Android.

Debido a que nuestra aplicación es un desarrollo de PFC, no está ubicada en Google Play, por lo que el procedimiento para su instalación dista un poco de las aplicaciones generales.

Por tanto lo primero que debemos hacer es conectar el dispositivo móvil al ordenador y pasar el fichero de la aplicación Eupct.apk a su tarjeta SD.

Posteriormente deberemos indicar al sistema operativo Android que debe permitir aplicaciones desde orígenes que no sean Google Play. Para ello activamos “Orígenes desconocidos” en Menú “Ajustes” => “Seguridad”:



Figura 4.1 Habilitación Orígenes Desconocidos

Por último deberemos mediante una herramienta de gestión de archivos de la memoria SD instalar nuestra aplicación. Existen diversas herramientas que realizan esta función en Google Play, nosotros hemos usado ASTRO File Manager:

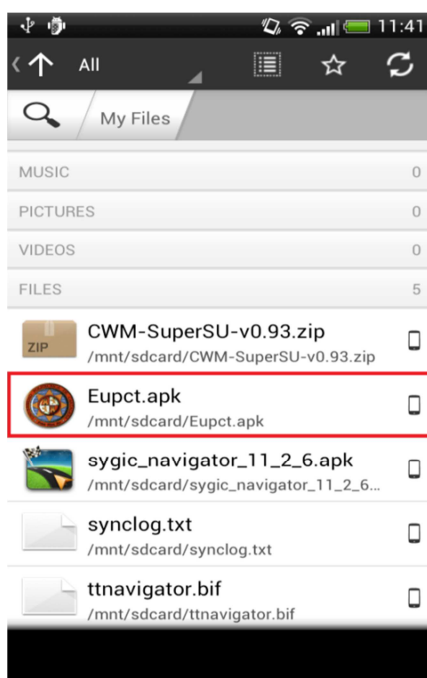


Figura 4.2 Vista Tarjeta SD paquete Eupct.apk



Buscamos el fichero, lo seleccionamos y automáticamente procederá a la instalación de la herramienta.

Si por el contrario, lo que quisiéramos es desinstalar la aplicación, sólo deberemos pulsar Desinstalar en:

Menú Ajustes => Aplicaciones => Señalamos Eupct



Figura 4.3 Desinstalar aplicación Eupct

4.1.2 Ejecución de la Aplicación. Casos de Uso.

Una vez seguidos los pasos del apartado anterior para instalar la aplicación Eupct, sólo nos queda pulsar sobre el icono de la UPCT que tendremos en el menú para lanzar la aplicación.

4.1.3 Funcionalidades Generales



A continuación vamos a describir funcionalidades y ejemplos de uso comunes a todos los usuarios de la aplicación, independientemente de los permisos que tengan (salvo los que no tengan permisos ninguno, que sólo verán los apartados del [5.2.1.1](#) a [5.2.1.3](#)).

4.1.3.1 Presentación

La bienvenida a la aplicación nos la da una imagen de la Universidad Politécnica de Cartagena que dura aproximadamente unos 4 segundos. El usuario sólo puede visualizarla, ya que no tiene interacción ninguna sobre ella.



Figura 4.4 Pantalla Bienvenida Aplicación

4.1.3.2 Acceso a la Aplicación por Usuario y Contraseña

Tras la pantalla de presentación, la aplicación nos redirige directamente a la pantalla de Acceso. Para acceder sólo deberemos indicar nuestro usuario (en este caso cuenta de la UPCT) y nuestra contraseña y pulsar el botón “Acceso”.

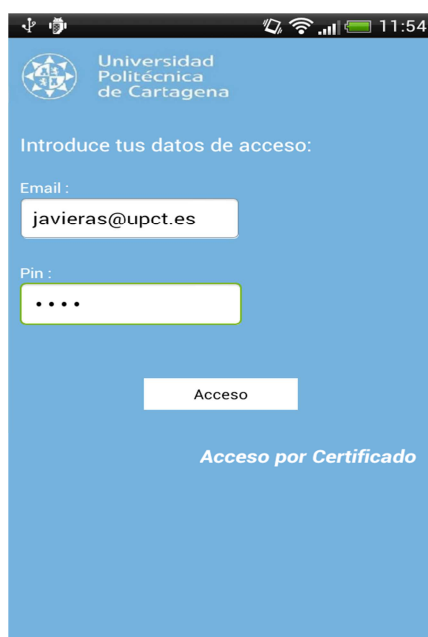


Figura 4.5 Pantalla Acceso

4.1.3.3 Acceso con Certificado Digital

Una de las posibilidades avanzadas de esta aplicación, es el acceso a ella a través de un certificado digital instalado en nuestro Android. Para ello, en vez de poner nuestro usuario y contraseña como en el apartado anterior, simplemente debemos pulsar sobre el texto secundario “Acceso por Certificado”



Figura 4.6 Pantalla Acceso con Certificado



Se nos mostrará un listado de los certificados instalados en nuestro teléfono. Si no tuviéramos ninguno instalado Android nos deja la posibilidad de instalar uno en esa misma pantalla. Seleccionamos el certificado que queremos usar y pulsamos sobre Permitir.

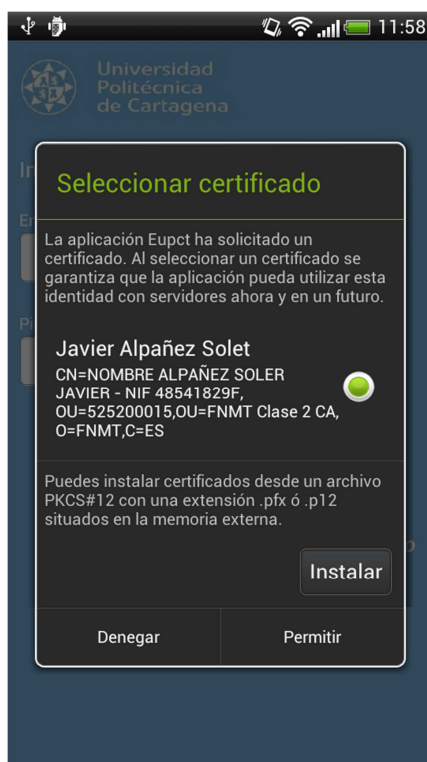


Figura 4.7 Selección de Certificado

Nota: Para instalar un nuevo certificado deberemos tener en nuestra tarjeta SD un fichero de almacenamiento de claves PKCS12. No debemos preocuparnos por la seguridad ya que después de añadirlo al repositorio de certificados propio de Android, éste elimina dicho fichero.

4.1.3.4 Navegación por los Espacios Colaborativos

Tras identificarnos en la aplicación, ésta nos lleva al listado principal de espacios de colaboración en los cuales tenemos algún permiso (de miembro o de administrador). Dichos espacios pueden ser, en el ámbito universitario, la carpeta de una asignatura, de una titulación, de un grupo de investigación, de un proyecto fin de grado, etc.



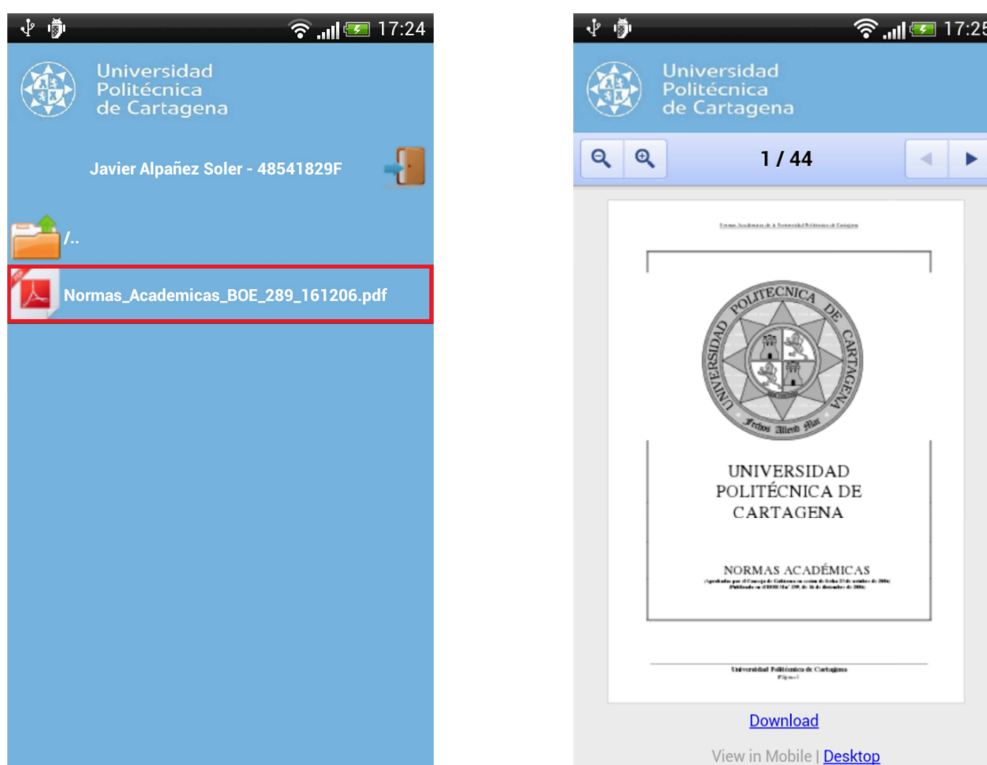
Visualizaremos el listado de ficheros y subcarpetas de cada carpeta seleccionada. Se puede subir y bajar de nivel, realizando una navegación por todos los subespacios. Para ello simplemente debemos hacer Click en la carpeta sobre la que queramos navegar.



Figura 4.8 Navegación por los Espacios Disponibles

4.1.3.5 Visualización de Ficheros

Además de poder pinchar sobre carpetas y navegar, podemos realizar click sobre cualquier fichero que veamos y automáticamente se nos mostrará. Para ello la aplicación nos llevará a la herramienta online de Google Docs.

**Figura 4.9 Visualización de Ficheros**

4.1.3.6 Visualización del Fichero de Firma Digital

Podemos observar en el listado, que pueden existir documentos cuyo icono identificativo contenga un candado. Eso significa que el administrador que ha subido a ese espacio el documento ha decidido darle mayor confianza al consumidor firmándolo digitalmente. Por tanto existe un fichero XML adjunto a ese fichero que contiene la firma y que no podemos ver en el listado.

Para verlo, mantenemos pulsado el documento que queremos ver la firma y se mostrará un submenú. Entonces señalamos “Ver Firma” y la aplicación nos llevará de nuevo a Google Docs para ver éste fichero XML con la firma digital en formato XADES.

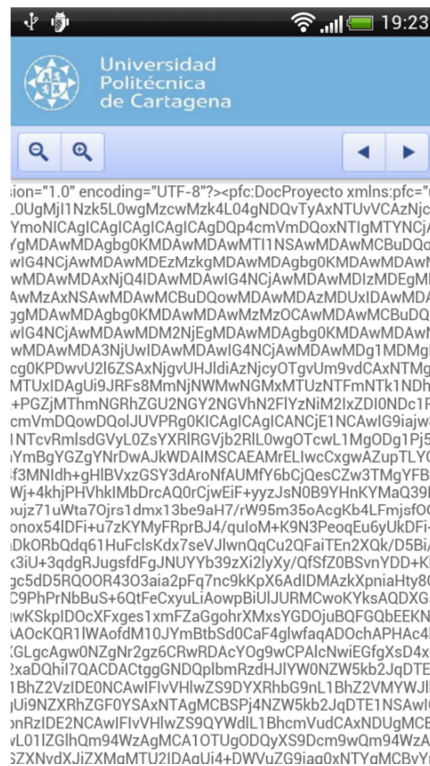


Figura 4.10 Visualización del Fichero de Firma Digital



4.1.3.7 Descarga de Ficheros a Tarjeta SD

Si el usuario no quisiera visualizar el documento sólo, sino que además prefiera guardarlo en la tarjeta SD, tiene dos opciones disponibles a través de la aplicación.

- A través de la aplicación.

La aplicación, al mantener pulsado el fichero a descargar, muestra un menú contextual que permite la opción de descargar. Para ello seleccionar “Guardar”. El fichero pasará a estar disponible en la tarjeta SD del usuario, exactamente en la ruta:

`\Android\data\es.upct.pfc.eupct\files`

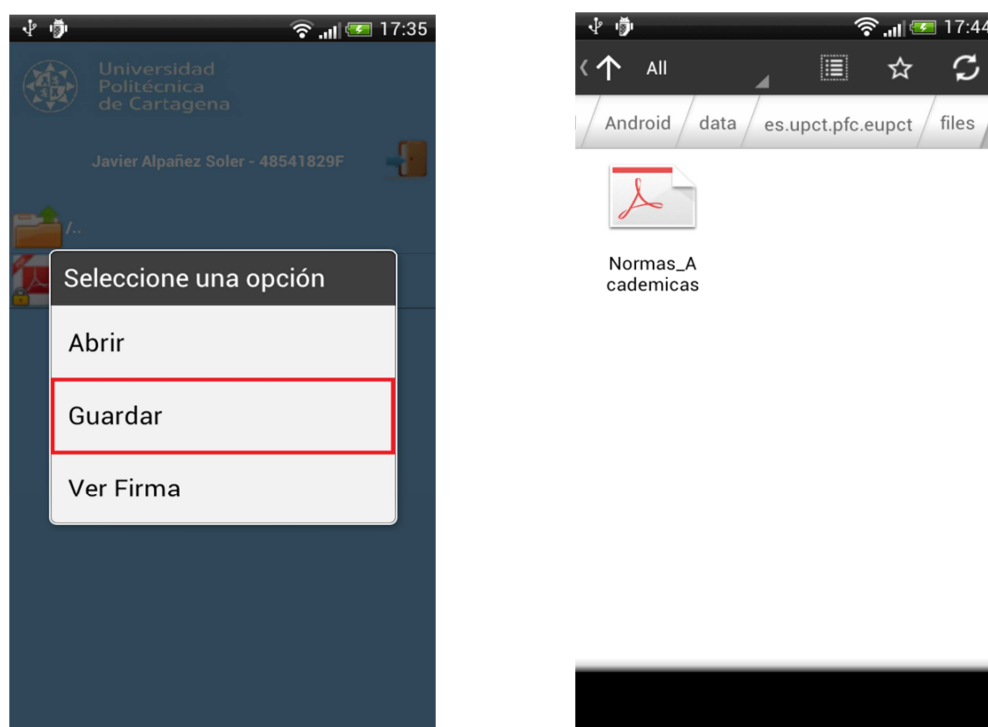


Figura 4.11 Descarga de Ficheros a Tarjeta SD a través de la aplicación

- A través de Google Docs

Mientras lo estamos visualizando online, en la parte inferior existe la posibilidad de descargar el fichero también. El lugar donde se guarda dependerá del navegador usado en nuestro Android, pero lo normal es que la ruta donde se guarde sea en la tarjeta SD `:\download`

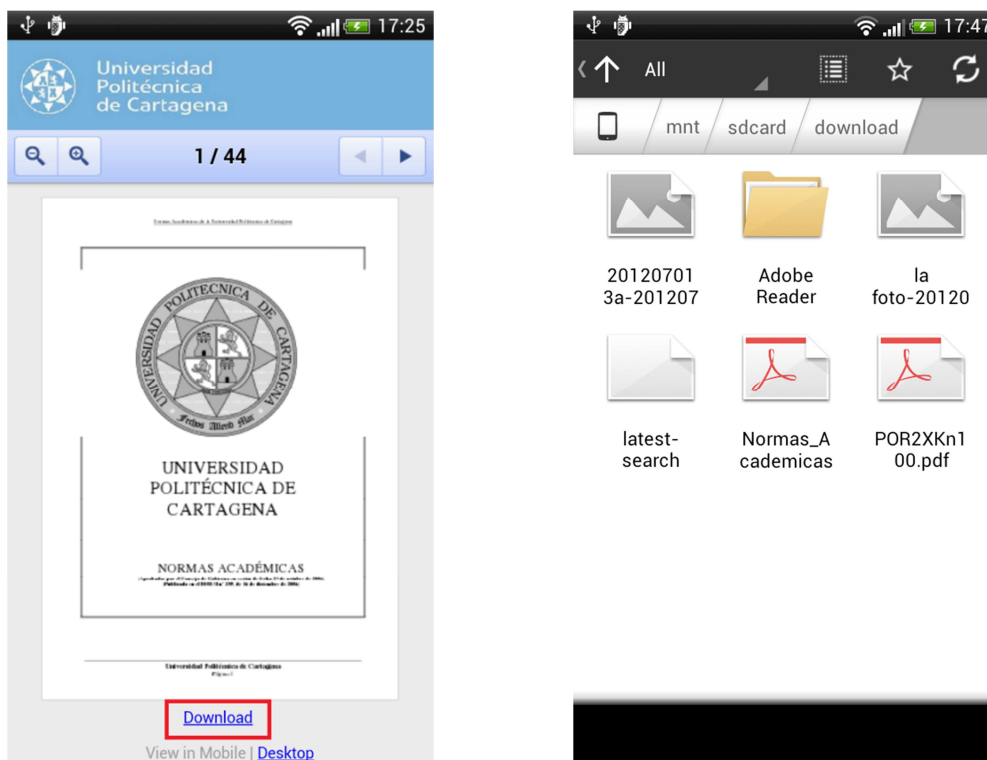


Figura 4.12 Descarga de Ficheros a Tarjeta SD a través de Google Docs

4.1.3.8 Notificaciones

Puede suceder que conforme estamos trabajando en la Aplicación o mirando otras aplicaciones en el móvil, tengamos una notificación de nuevo documento. Esto quiere decir que algún administrador de un espacio en los cuales tenemos permisos de acceso ha subido un documento nuevo.

Por tanto la aplicación nos avisará con una (o más) notificación emergente en la parte superior que tenemos un nuevo documento disponible. Si desplegamos la parte superior obtendremos más información. Además, si pulsamos sobre ella, la aplicación Eupct se nos mostrará en primer plano y veremos la carpeta que contiene dicho fichero, por si queremos verlo, descargarlo, etc.



Figura 4.13 Notificaciones



4.1.3.9 Menú Información

Si un usuario despliega el menú de la aplicación pulsando el botón “MENU” del teléfono. Según el modelo de móvil, este botón puede ser táctil o no, tener un dibujo u otro...

Una vez vemos el menú de información, si pulsamos sobre él se nos mostrará la información de la aplicación. Esta información podría ser interesante si quisiéramos mostrar por ejemplo un texto de Ayuda sobre la pantalla actual.

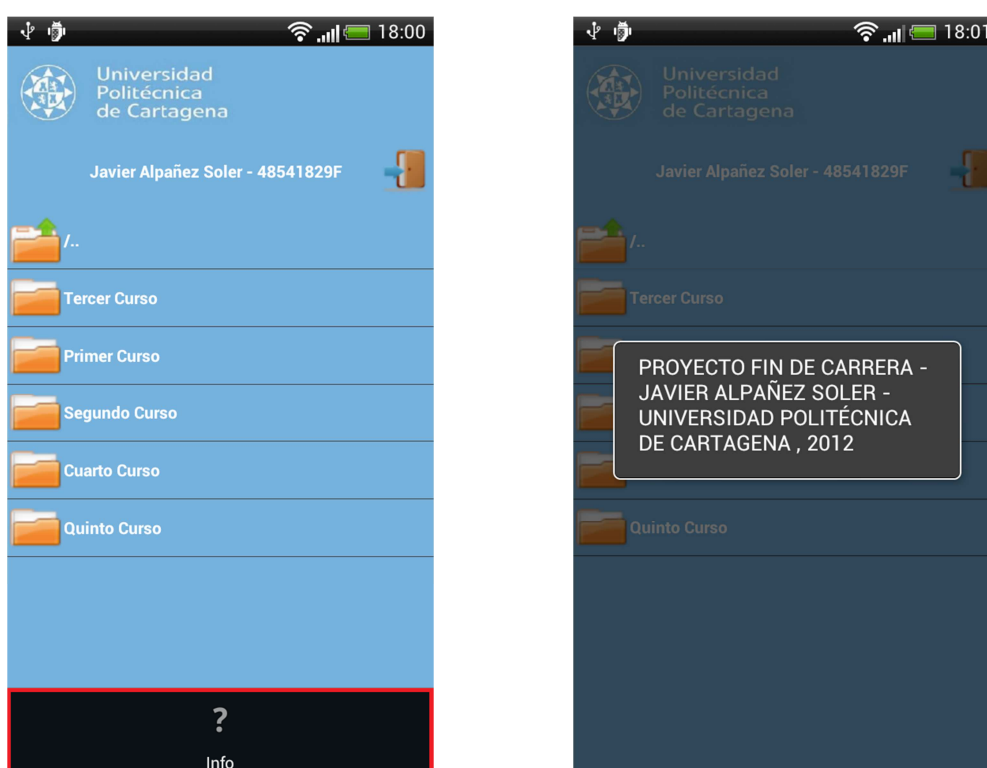


Figura 4.14 Ventana Información

4.1.3.10 Salir de la Aplicación

Es posible realizar el ‘logout’ o salida de la aplicación mientras el icono superior derecho habilitado para ello. Esto permitirá, entre otras cosas, dejar de recibir notificaciones para documentos nuevos.

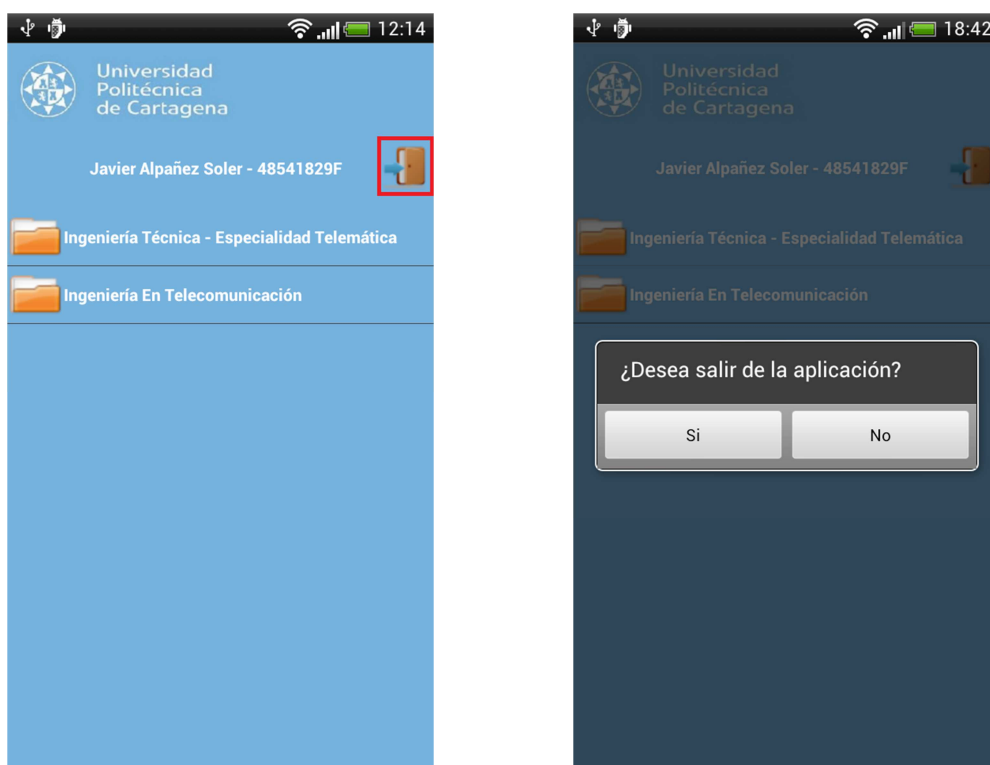


Figura 4.15 Salir de la Aplicación

4.1.4 Funcionalidades sólo para Administradores

Existen acciones en la aplicación que sólo pueden realizar los administradores de los espacios (y sus subcarpetas). A continuación definimos dichas acciones:

4.1.4.1 Creación Carpetas

Cuando un administrador quiere crear una nueva carpeta en el espacio compartido y que todos los miembros lo vean, sólo tendrá que desplegar dicha opción con el botón “MENÚ” del teléfono.

Una vez seleccionado, se nos mostrará la opción de crear la carpeta mostrándonos una nueva ventana para introducir el nombre de la carpeta. Tras pulsar en “Aceptar”, la aplicación recargará el listado visualizado y ya tendremos nuestra carpeta creada. Por ejemplo para crear la carpeta “Prácticas”:

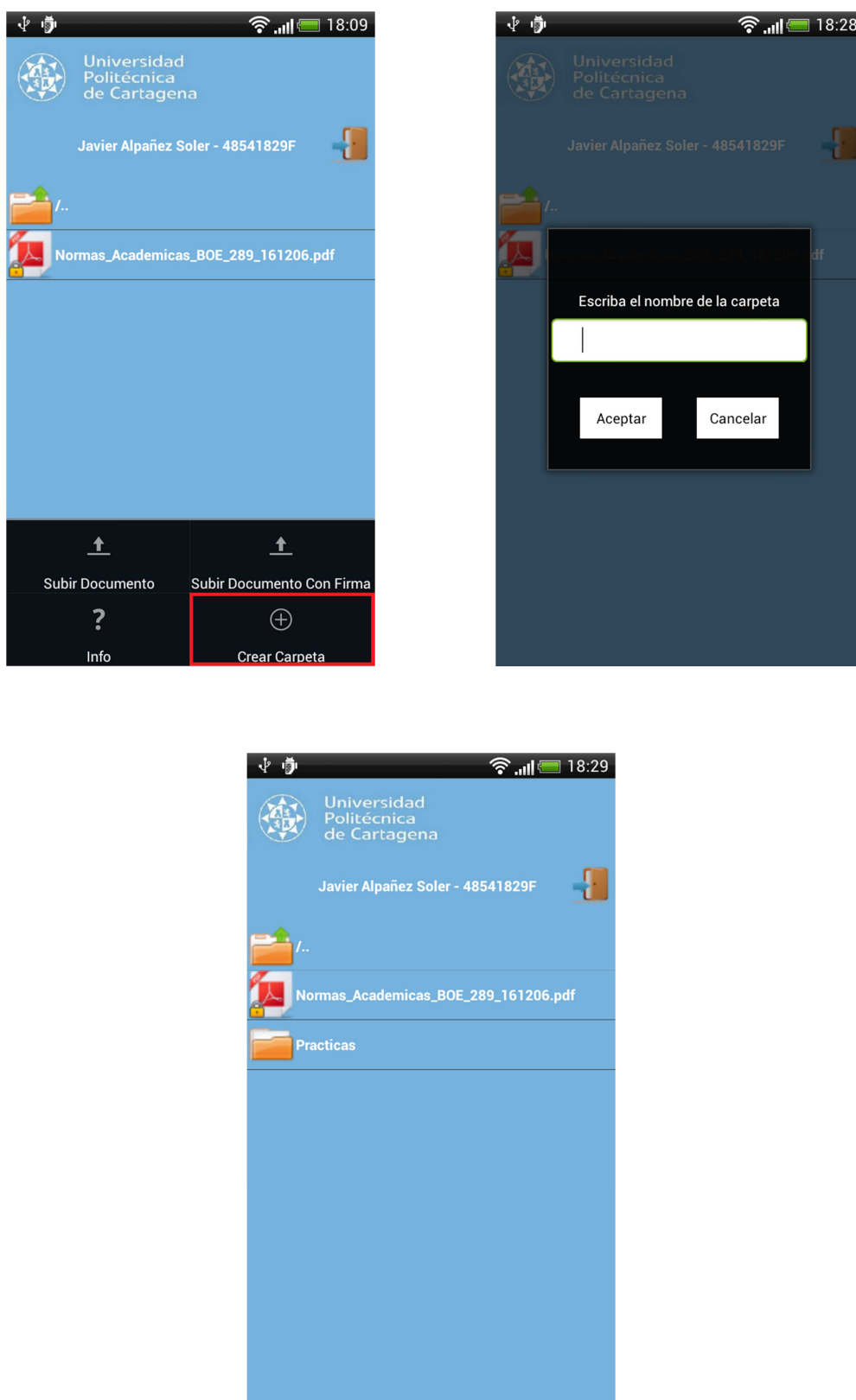


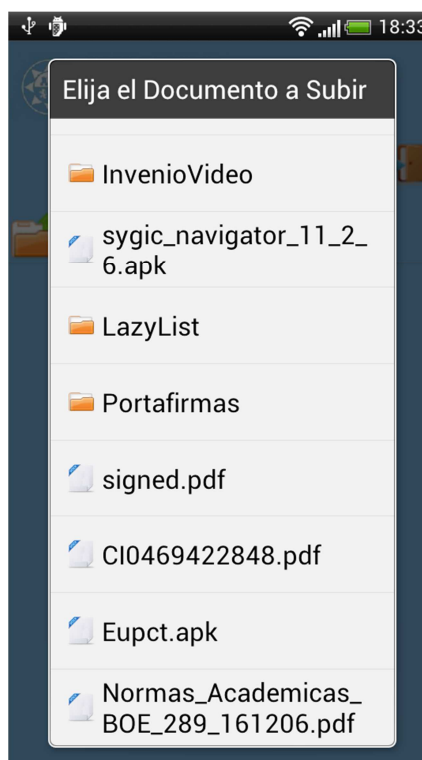
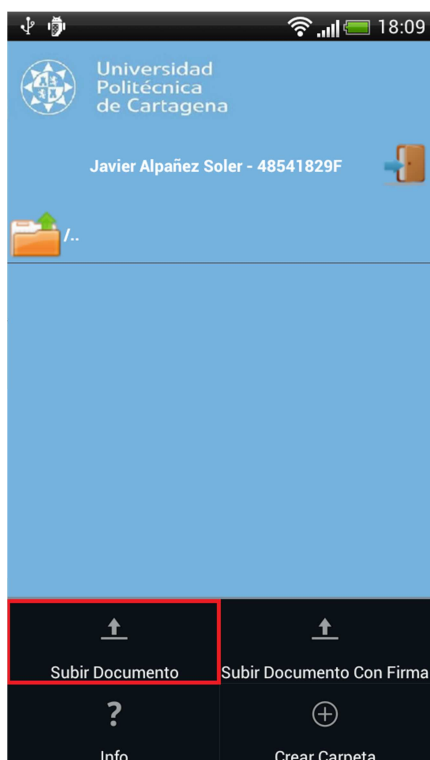
Figura 4.16 Crear una Carpeta



4.1.4.2 Subida Ficheros

Si lo que un administrador quiere es subir un fichero a la carpeta que está visualizando en ese momento, sólo deberá pulsar el botón “MENU” del teléfono y seleccionar la opción “Subir Documento”.

Después se nos mostrará una ventana con los ficheros de nuestra tarjeta SD. Seleccionamos el fichero que queremos. La aplicación procederá a subirlo y una vez terminado nos lo notificará y recargará la carpeta donde podremos ver que se ha subido correctamente.



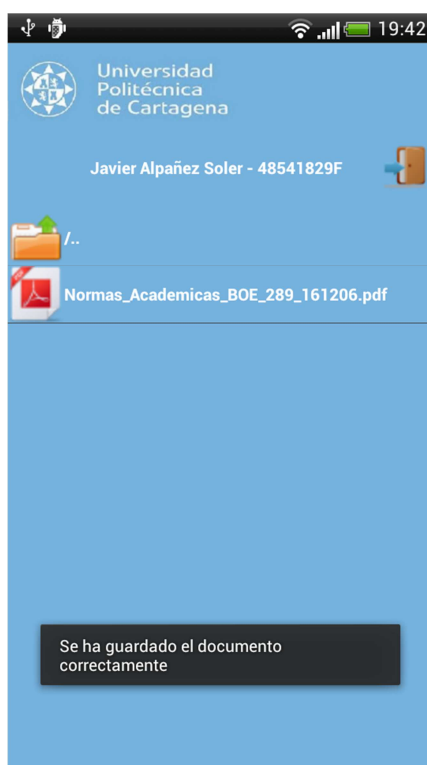


Figura 4.17 Subir un Fichero

4.1.4.3 Subida Y Firma de Ficheros

Es posible que un administrador que quiere subir un documento según se explica en el apartado [5.2.2.2](#) , además quiera darle una mayor consistencia y seguridad firmando el documento. Para ello debemos pulsar el botón “MENU” del teléfono y marcar “Subir Documento con Firma”.

Se nos abrirá una ventana con los ficheros de nuestra tarjeta SD para que seleccionemos el fichero correspondiente. Una vez marcado, se nos abrirá el repositorio de certificados instalados en el móvil. Si no tuviéramos ninguno instalado, nos dejará la opción de instalar uno en ese momento.

Una vez marcado un certificado, la aplicación realizará la subida y firma del fichero deseado. Tras ello notificará el resultado (si es correcto o no el proceso) y recargará de nuevo la carpeta. Si todo ha ido correctamente deberíamos ver el fichero subido, junto con un icono con un candado que indica que ha sido firmado y que contiene un fichero XML asociado con la estructura XADES.

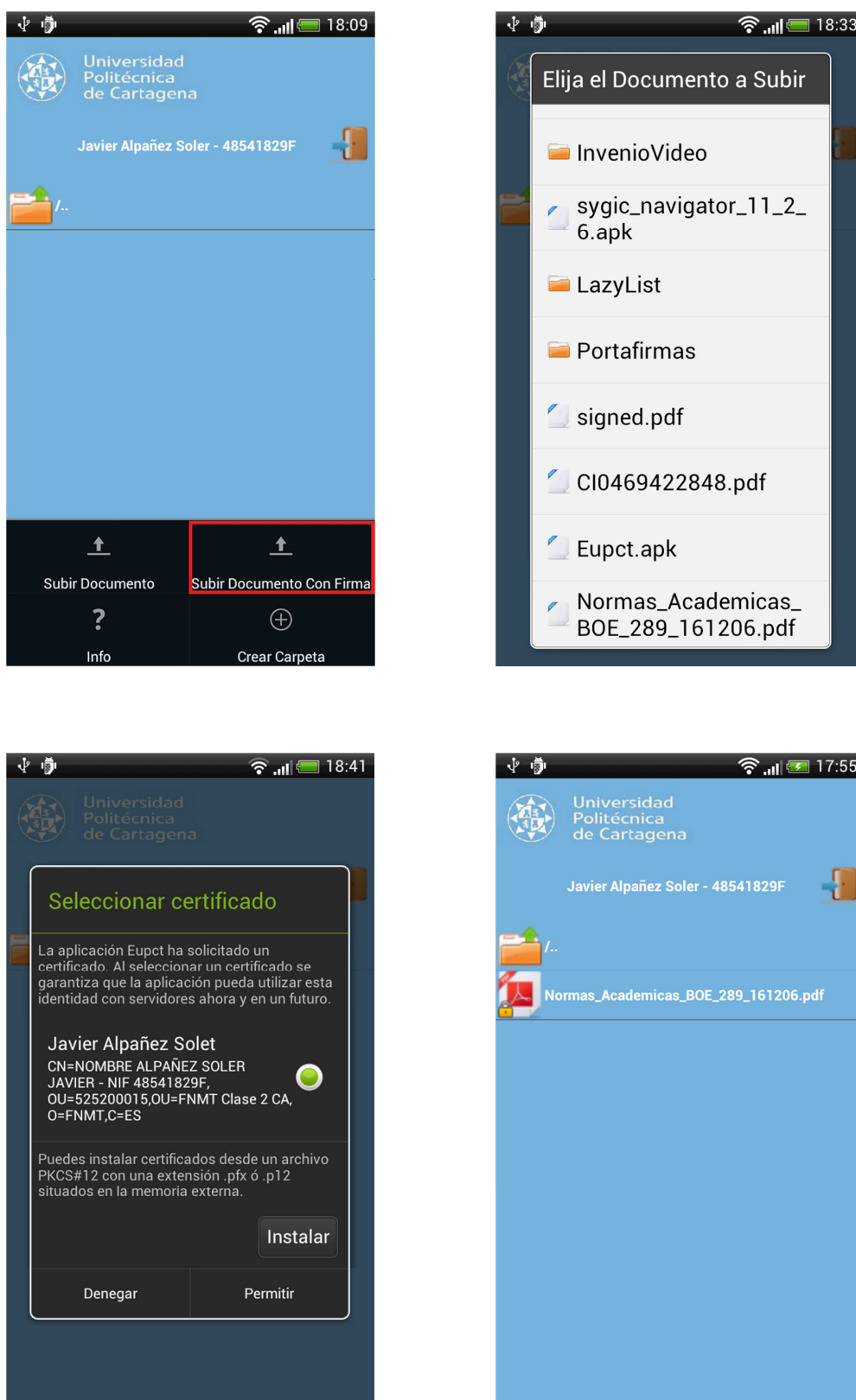


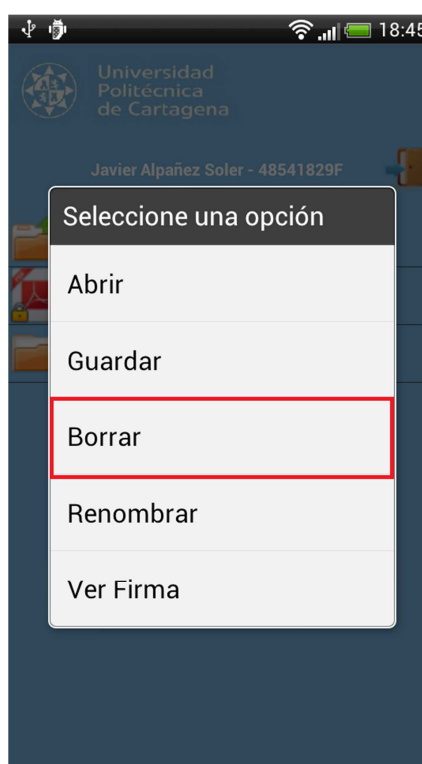
Figura 4.18 Subir y Firmar un Fichero



4.1.4.4 Borrar Ficheros / Carpetas

Cuando un Administrador quiere borrar un fichero o carpeta que está viendo, es posible hacerlo con unos simples pasos.

Sobre la carpeta o fichero correspondiente deberemos pulsar y no soltar, lo que nos mostrará un menú de opciones. Podemos ver la opción “Borrar”. Tras pulsar, observamos que ya no existe el documento o carpeta correspondiente. En el ejemplo vamos a mostrar como borrar un fichero anteriormente subido:



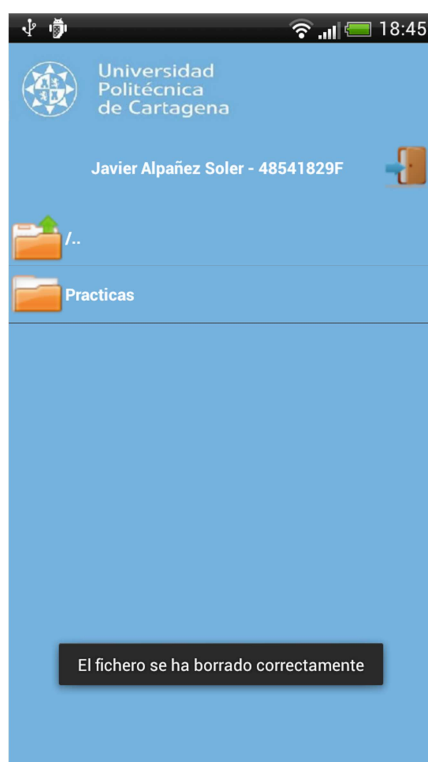


Figura 4.19 Borrado de Carpeta / Fichero

4.1.4.5 Renombrado de Ficheros / Carpetas

Si queremos renombrar una carpeta o fichero, deberemos realizar los siguientes pasos.

Sobre la carpeta o fichero correspondiente deberemos pulsar y no soltar, lo que nos mostrará un menú de opciones. Podemos ver la opción “Renombrar”. Tras pulsar, observamos que se nos abre una nueva ventana para introducir el nuevo nombre del fichero o carpeta. Una vez introducido sólo debemos pulsar aceptar y el fichero o carpeta habrá cambiado de nombre si todo ha ido correctamente. Para el ejemplo vamos a mostrar cómo renombrar una carpeta:

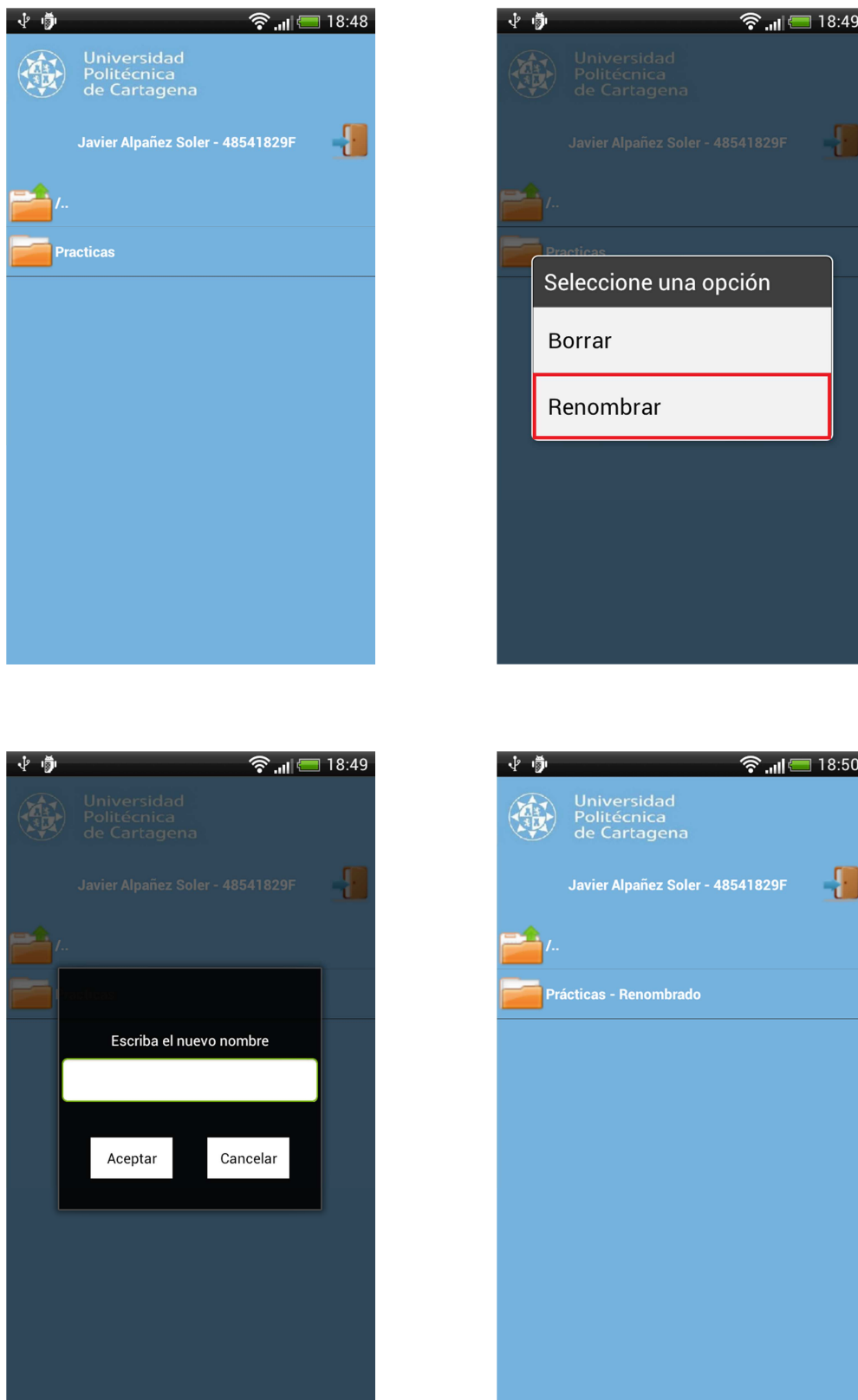


Figura 4.20 Renombrado de Carpeta / Fichero



4.2 Administración Servidor

Para explicar cómo administrar el servidor, necesitaremos tener algunos conocimientos previos sobre bases de datos. En cuanto al gestor documental no son necesarios ya que su administración web es bastante sencilla e intuitiva.

4.2.1 Ejecución de los Servicios. Casos de Uso.

Una vez arrancados todos los servicios de la parte del servidor tal como se explica en los apartados [4.4.1.1](#), [4.4.2.2](#) y [4.4.3.2](#) de este documento, pasamos a ver qué acciones de administración básicas serán necesarias para el correcto funcionamiento.

4.2.2 Alta de un Usuario.

Una funcionalidad muy común, es dar de alta un usuario en la plataforma para que pueda identificarse en la aplicación móvil.

Para ello deberemos abrir el administrador de base de datos MySQL que más nos guste (nosotros usamos ToadforMySQL) y usar y realizar los siguientes pasos.

Primero nos debemos conectar con la base de datos incluyendo la url y los credenciales correctamente, en nuestro caso:

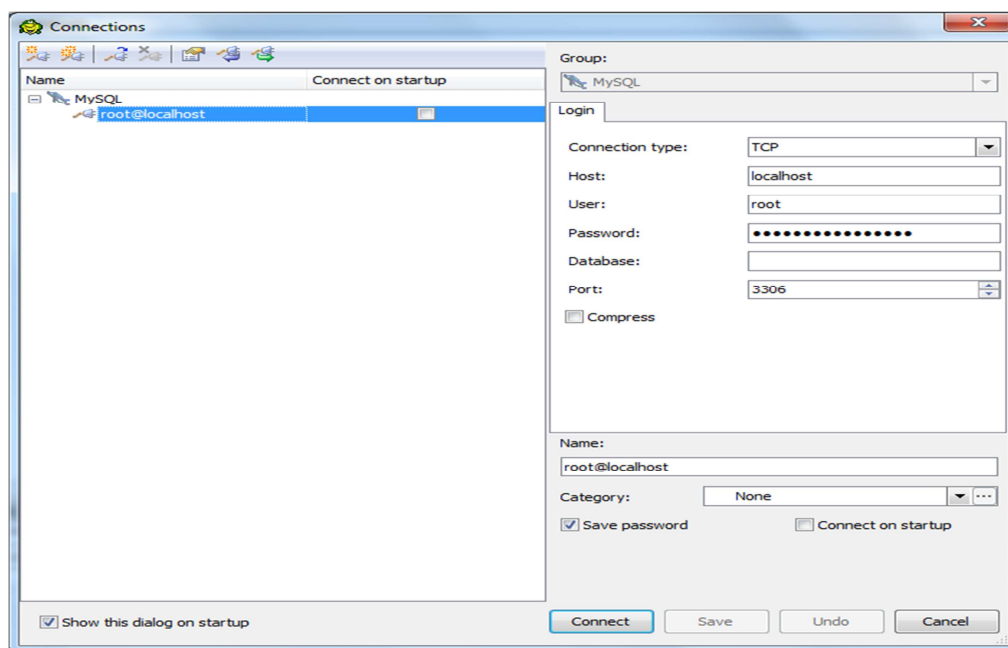


Figura 4.21 Acceso a Base de Datos con ToadForMySQL



Nota: Cómo ya se dijo con anterioridad, recordamos que los credenciales son para user :root y para password : (vacía)

Una vez identificados, nos vamos a la carpeta “usuarios” y veremos un listado de los usuarios ya activos en la plataforma:

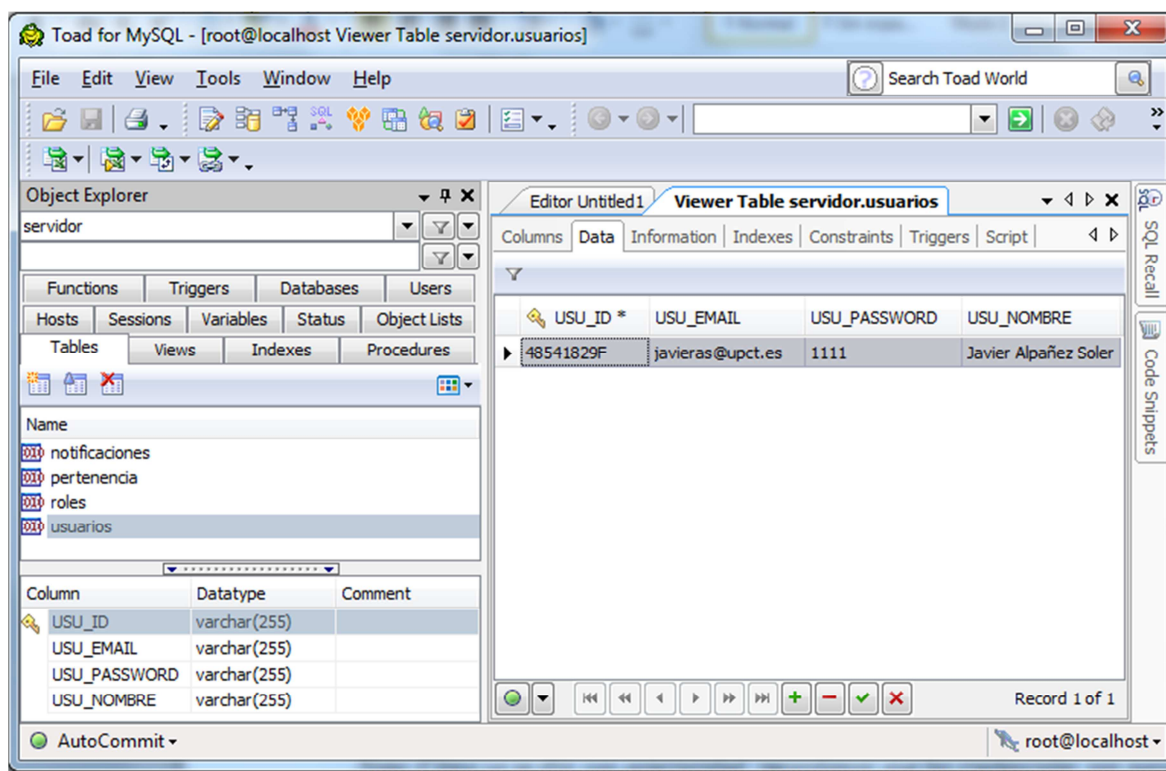


Figura 4.22 Tabla Usuarios Base de Datos

Para insertar en esa tabla un nuevo usuario, que es lo que buscamos, simplemente nos abrimos una pestaña de edición de código SQL y añadimos lo siguiente, cambiando estos valores de ejemplo por los que queramos usar. Posteriormente le damos a ejecutar SQL:

```
insertinto usuarios(USU_ID      ,USU_EMAIL      ,USU_PASSWORD      ,USU_NOMBRE)
VALUES('12345678A','nuevousuario@upct.es','contraseña','Nuevo Usuario')
```

Cuadro 4.1 Añadir un Nuevo Usuario



Recordar que el USU_ID, que es el NIF o DNI de la persona que queremos dar de alta, no puede estar anteriormente insertada en la tabla. Una vez realizado volvemos a la vista de la tabla usuarios y veremos nuestro registro insertado:

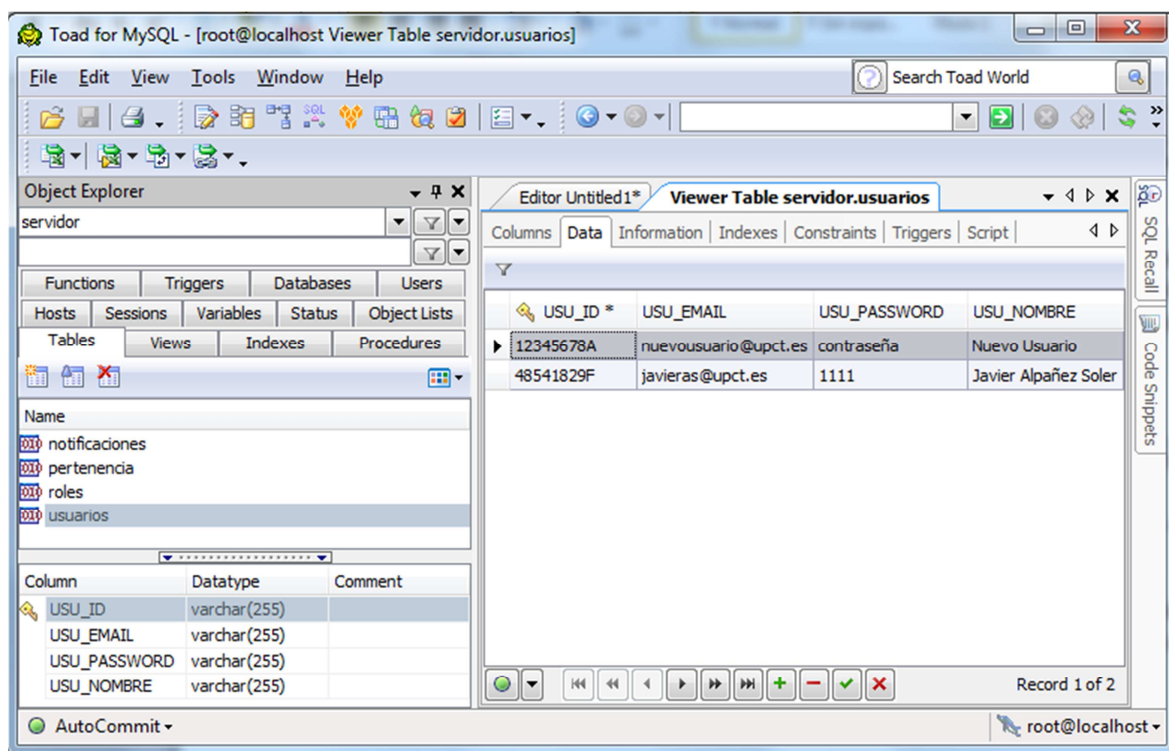


Figura 4.23 Tabla de Usuarios con el Nuevo Usuario

4.2.3 Creación de un nuevo Espacio de Colaboración

Otra de los requerimientos más comunes, es dar de alta un nuevo espacio para que varios usuarios compartan documentos en la aplicación. Esta acción se lleva íntegramente en el gestor documental Alfresco.

Primero accedemos en cualquier navegador a la pantalla de administración, en nuestro caso la url es:

```
http://79.109.182.20:7777/alfresco
```

Cuadro 4.2 URL Administrador Web Alfresco



Y veremos la pantalla de acceso, que tiene este aspecto :



The login screen features the Alfresco logo at the top. Below it, the text 'Introducir datos de inicio de sesión:' is displayed. There are three input fields: 'Nombre de usuario:', 'Contraseña:', and 'Idioma:' (with a dropdown menu showing 'Spanish'). A 'Iniciar sesión' button is located at the bottom right of the form.

Figura 4.24 Pantalla Acceso Alfresco

Nota: Cómo ya se dijo con anterioridad, recordamos que los credenciales son para nombre de usuario :admin y para contraseña : admin.

Una vez es correcto el acceso, se nos mostrará la ventana de bienvenida que tiene el siguiente aspecto:



The dashboard shows the 'Mi Panel de Inicio Alfresco' (My Alfresco Start Panel). It includes a navigation sidebar on the left with options like 'Mi Alfresco', 'Espacio de empresa', 'Mi Espacio', and 'Espacio de invitado'. The main content area is titled 'Guía de inicio' (Getting Started) and contains sections for 'Alfresco' (with links to Demostración, Panorama de funcionalidades, and Ayuda en línea) and 'Tareas comunes' (with links to Examinar elementos en su espacio personal, Crear un espacio en su espacio personal, Añadir elementos en su espacio personal, and Crear contenidos en su espacio personal). At the bottom, there is a section for 'Mis Tareas Pendientes' (My Pending Tasks) which currently shows 'Ninguna tarea encontrada.' (No tasks found).

Figura 4.25 Administración Alfresco



A continuación debemos desplegar en el menú de la izquierda la pestaña “Mi Espacio” y a continuación “Sitios”. Veremos los espacios que actualmente tenemos dados de alta. Para dar de alta uno nuevo, pulsamos en la parte superior derecha en “Crear” => “Crear Espacio” tal como se indica en la siguiente figura:

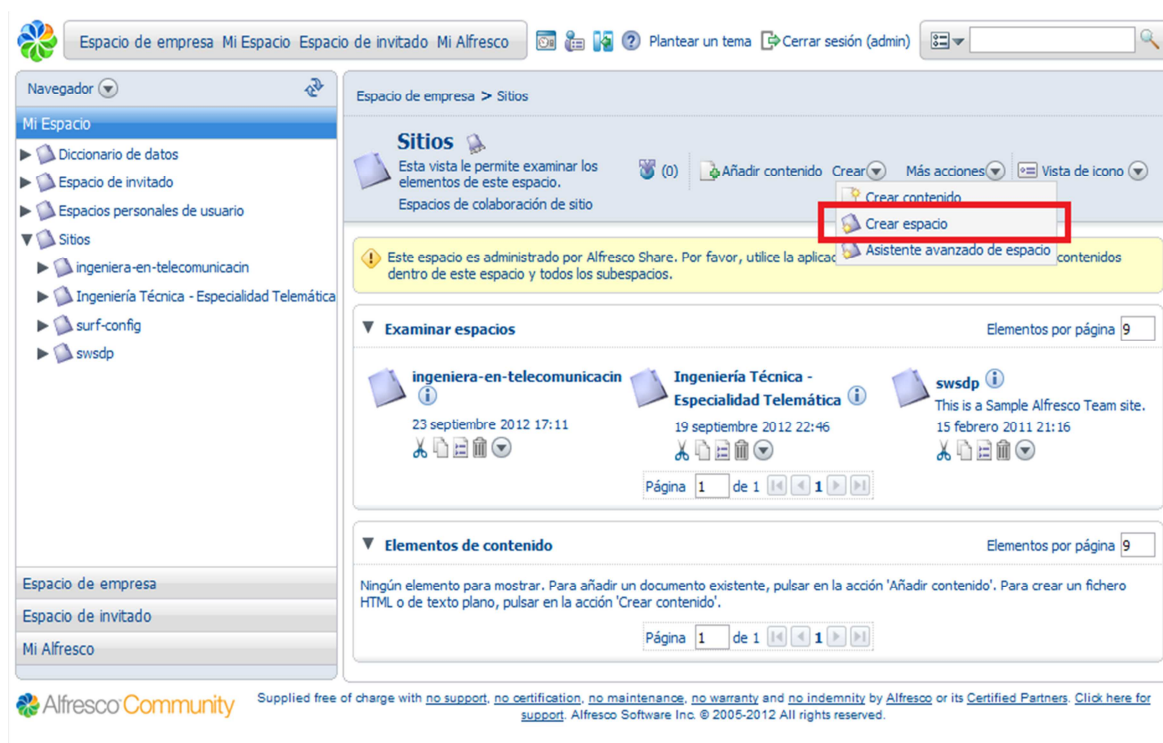


Figura 4.26 Crear un Espacio

Se nos abrirá una nueva ventana donde sólo deberemos indicar el nombre del nuevo espacio y pulsar en la parte derecha en “Crear espacio”:

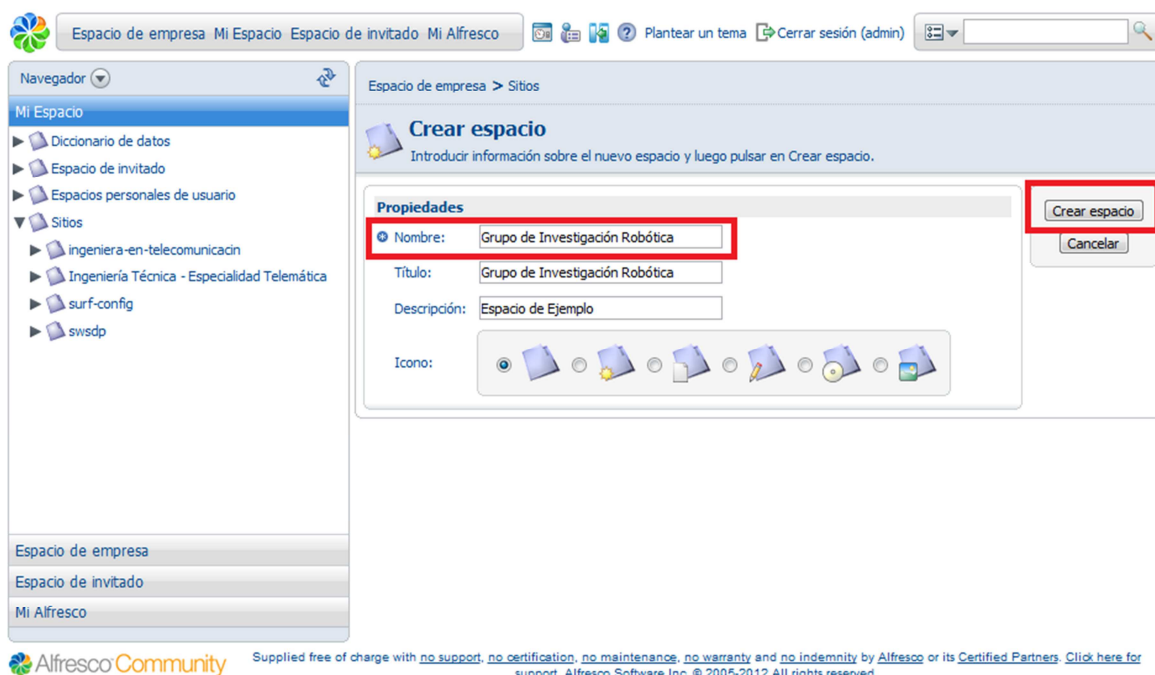


Figura 4.27 Añadir Nombre Nuevo Espacio

Observamos entonces que ya tenemos nuestro nuevo espacio creado junto al resto de espacios:

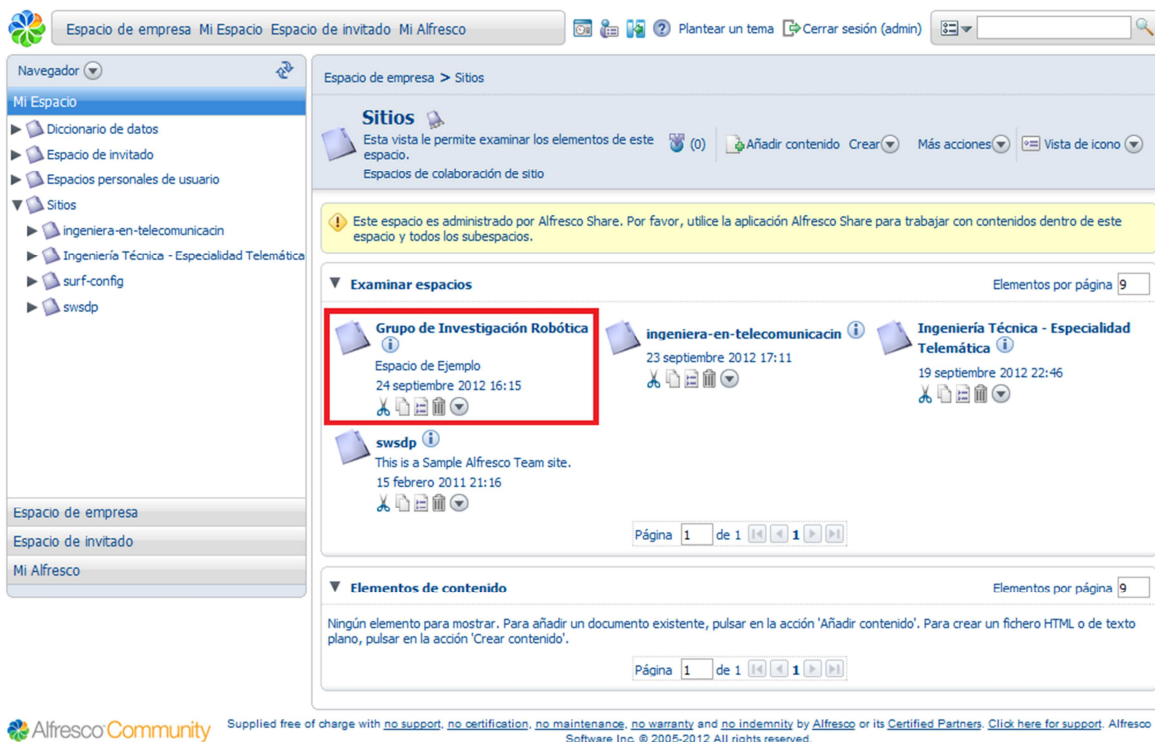


Figura 4.28 Listado Espacios Disponibles Actualizados



4.2.4 Asignación de Permisos a un Usuario en un Espacio

Una vez hemos visto en los pasos anteriores cómo crear un usuario nuevo y un nuevo espacio colaborativo, el flujo lógico nos llevaría a asignar el usuario a ese espacio con un rol determinado.

Para ello nos vamos a la pantalla donde está el espacio en el gestor documental (visto en el apartado anterior) y pulsamos sobre el icono “Ver los detalles”:

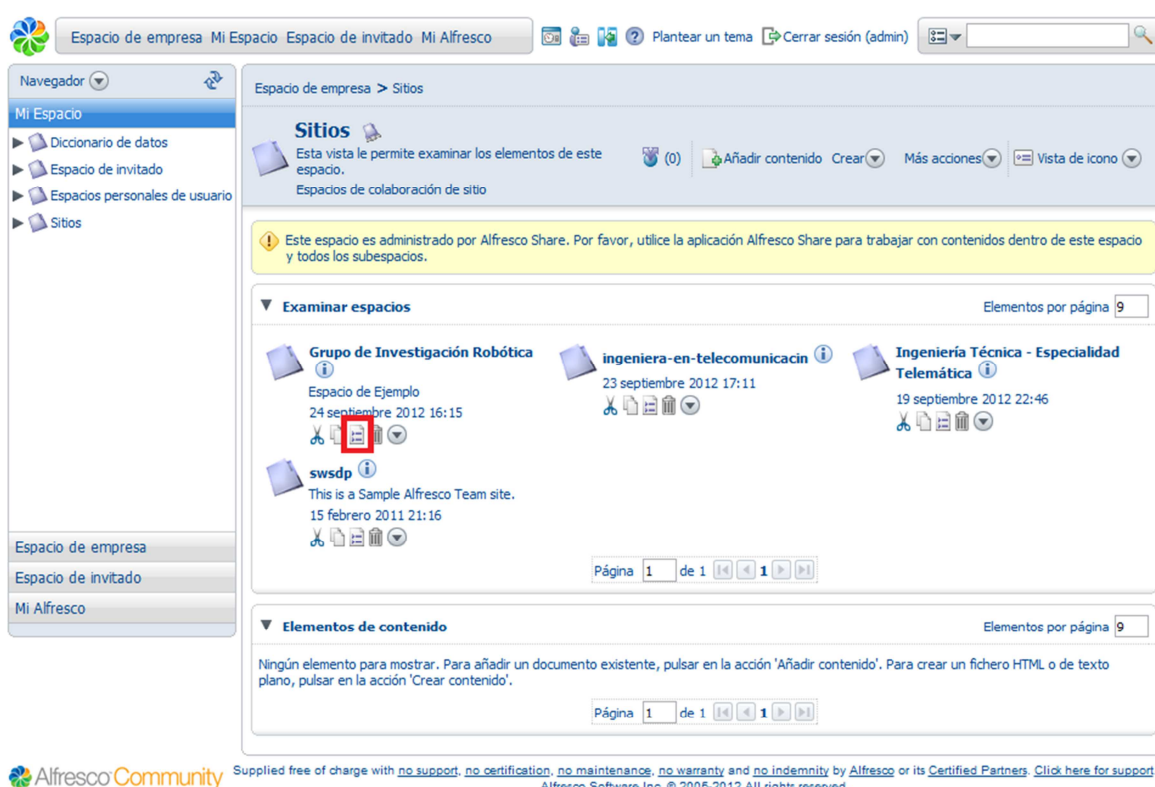


Figura 4.29 Ver Detalles de un Espacio

Nos aparecerá una página con los detalles del espacio. Necesitamos ver su UUID identificativo único. Una forma simple de usarlo es pulsar sobre “Referencia de nodo Alfresco”:

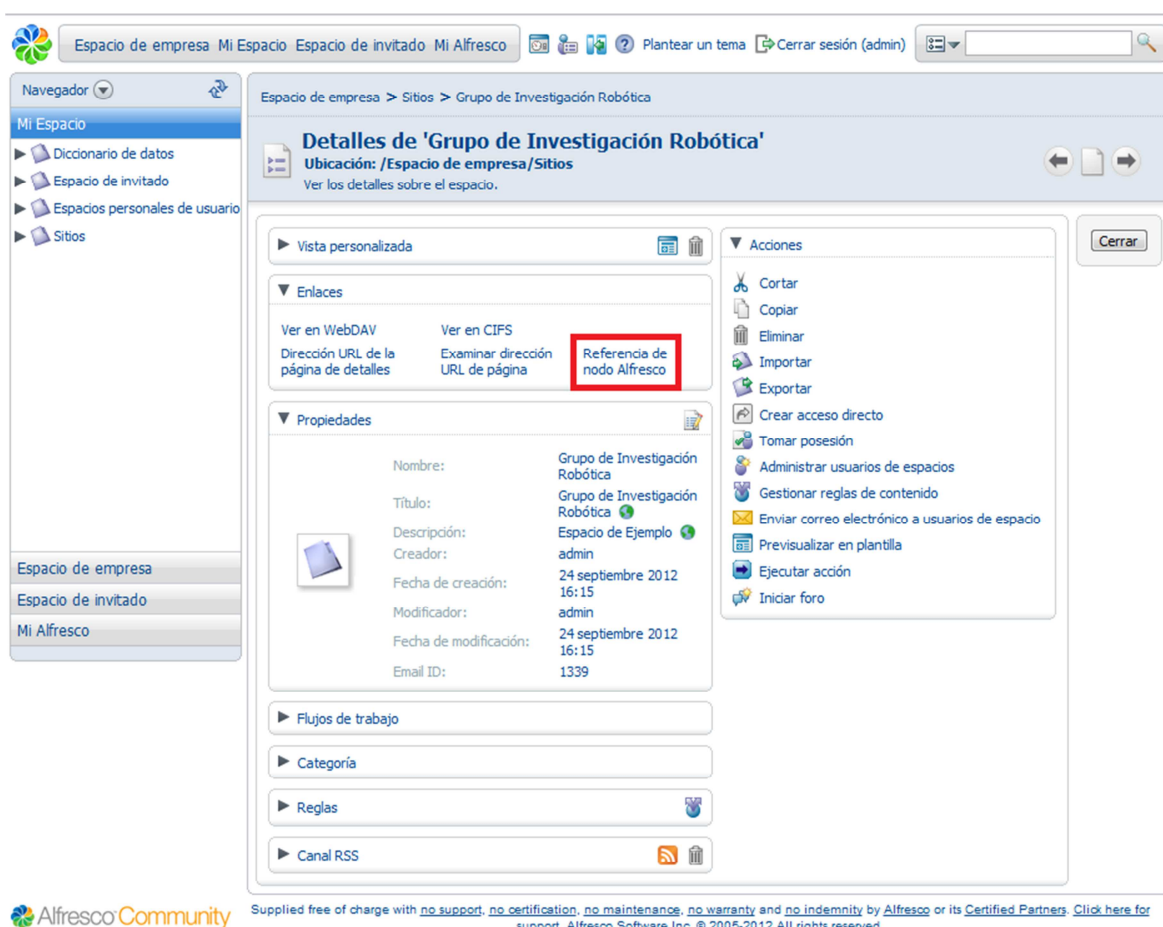


Figura 4.30 Recuperar UUID de un Espacio

Entonces se nos abrirá una ventana vacía con una dirección. En nuestro caso de ejemplo, la URL es:

```
workspace://SpacesStore/3b49b9d3-88c3-4cd8-843e-36258087f2c2
```

Cuadro 4.3 URL del Espacio Buscado

Y por tanto, el valor que buscamos es el UUID: “3b49b9d3-88c3-4cd8-843e-36258087f2c2”.

A continuación nos vamos al gestor de base de datos como se explica en el apartado [5.2.2](#) de este documento. Para realizar la asignación, vamos a insertar un registro en la tabla “pertenencia”. Abrimos una ventana de Editor de consulta SQL y escribimos lo siguiente (cambiar los valores del ejemplo por los que sean necesarios):



```
insert into pertenencia(PER_ID,USU_ID,PER_UUID,PER_NOMBRE,ROL_ID) VALUES  
(0,'12345678A','3b49b9d3-88c3-4cd8-843e-36258087f2c2','Grupo de Investigación  
Robótica',1)
```

Cuadro 4.4 Insertar Una Pertenencia a Un Espacio

Vamos a explicar los valores de cada campo, para que para otro caso de uso no haya dudas en su aplicación:

PER_ID: Es el identificador del registro en la tabla. Debe ser un valor incrementado cada vez. En nuestro caso lo ponemos manualmente y ponemos 0 ya que es el primer registro.

USU_ID: NIF o CIF del usuario. Recordamos que dicho usuario debe estar creado con anterioridad, sino no dejará guardar el usuario.

PER_UUID: Valor UUID del espacio a asignar al usuario. En nuestro caso es la cadena recogida antes del gestor documental.

PER_NOMBRE: Nombre del espacio que queremos que aparezca en la aplicación del cliente. No tiene por qué ser el mismo nombre que el que se le puso cuando lo creamos en el gestor documental.

ROL_ID: Permisos dentro de ese espacio. Las opciones posibles actualmente son: 1 administrador, 2 miembro. Se pueden consultar en la tabla “roles”.

Una vez lanzada la sentencia SQL, podremos ver nuestro registro creado en la tabla “pertenencia”.

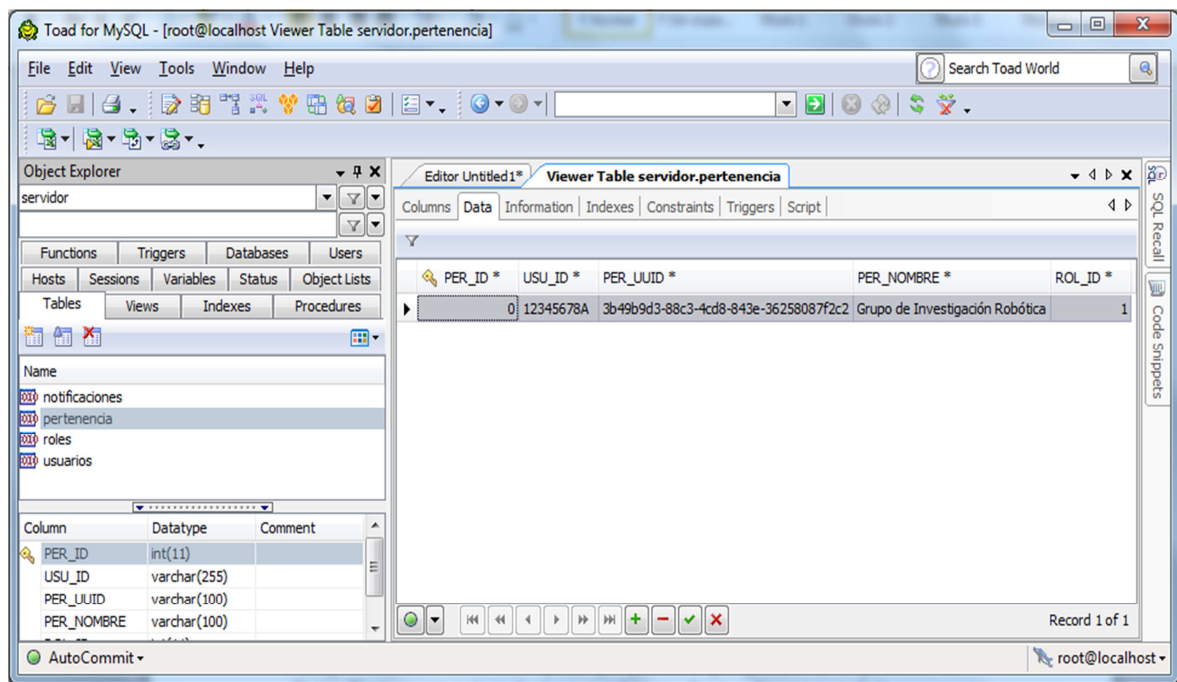


Figura 4.31 Vista Tabla Pertenencia

Como resultado final, vamos a comprobar que el usuario ve este espacio en su teléfono móvil tras identificarse y puede trabajar normalmente sobre él.



Figura 4.32 Vista en la Aplicación del Nuevo Espacio Creado



Capítulo 5

Conclusión y Líneas Futuras

El proyecto ha conseguido abarcar un gran número de tecnologías diversas y ampliamente utilizadas de tal forma que se ha conseguido finalmente una estructura bastante amplia sin que resulte complicado su mantenimiento.

Además, podemos añadir que se han alcanzado no sólo los objetivos marcados inicialmente, sino que además hemos podido realizar desarrollos avanzados que añadan una línea de investigación y dificultad adicional.

Se ha podido obtener una aplicación visualmente agradable y sencilla para el usuario. Sin embargo esto no ha sido fácil ya que para su implementación y funcionamiento tenemos detrás toda una cadena de tecnologías que para el desarrollador recién iniciado puede resultar muy complejo.

Por tanto la mayor dificultad del proyecto fue el estudio e implementación real de todas y cada una de estas tecnologías ya que no todas fueron explicadas durante la carrera.

Unas posibles líneas futuras de la aplicación para investigar y desarrollar sería:

- Ampliar el concepto de espacio colaborativo, no sólo a documentos de todo tipo, sino a interacción en tiempo real como por ejemplo chat entre los miembros del espacio, blogs de notas o tareas.... ya que actualmente no existe comunicación entre los participantes más allá de la parte documental.
- Posibilidad de que se puedan recibir noticias en el espacio. Imaginemos que un profesor quiere notificar el cambio de aula o laboratorio a la hora de impartir una práctica, etc.
- Pasar a servicios web de 2º generación tipo REST ya que es capaz de soportar una mayor carga y es bastante más estable que el SOAP implementado para millones de llamadas.
- Aprovechar el punto iniciado en cuanto a certificado digital y firma digital y realizar comprobación de la firma XADES generada.
- En la línea de lo anterior, debido que tenemos ya el acceso al certificado listo, podemos cifrar la comunicación para hacerla más segura por HTTPS por ejemplo.



- Realizar una interfaz, en la aplicación Android o web, para la gestión del servidor y su base de datos, ya que ahora se realiza con frameworks externos como ToadForMySQL.
- Utilizar el directorio de usuarios de la UPCT basado en LDAP y realizar la comprobación de usuarios respecto a él.
- Ampliación de los roles soportados. Además del miembro y administrador, también se podría tener algún invitado (sólo lectura), editor (encargado de subir sólo documentos), etc.
- Generación de reglas de asignación dinámica de permisos. Podemos conectar con LDAP y asignar una regla que todo alumno matriculado en una facultad, automáticamente tenga acceso al espacio de esta facultad.



Capítulo 6

Bibliografía

A continuación listamos un conjunto de direcciones web y documentación utilizada para informarnos en el desarrollo del proyecto.

- <http://www.xatakandroid.com/sistema-operativo/que-es-android>
- <http://es.wikipedia.org/wiki/Android>
- <http://developer.android.com/index.html>
- <http://www.sgoliver.net/blog/?p=1278>
- <http://es.wikipedia.org/wiki/Alfresco>
- <http://www.alfresco.com/es/>
- http://es.wikipedia.org/wiki/Firma_digital
- http://es.wikipedia.org/wiki/Firma_XML
- <http://oficinavirtual.mityc.es/componentes/MITyCLibXADES/integrator-guide/detached-sign.html>
- <http://es.wikipedia.org/wiki/MySQL>
- http://es.wikipedia.org/wiki/Servicio_web
- <http://es.wikipedia.org/wiki/SOAP>
- <http://es.wikipedia.org/wiki/Hibernate>
- [http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))
- http://es.wikipedia.org/wiki/Desarrollo_de_Programas_para_Android
- <http://developer.android.com/sdk/index.html>
- <http://es.wikipedia.org/wiki/Toad>